

---

---

**Industrial automation systems and  
integration — Product data  
representation and exchange —**

Part 223:

**Application protocol: Exchange of design  
and manufacturing product information  
for cast parts**

*Systèmes d'automatisation industrielle — Représentation et échange  
de données de produits —*

*Partie 223: Échange de dessin et d'information sur la conception et la  
fabrication de pièces de fonderie*



Reference number  
ISO 10303-223:2008(E)

© ISO 2008

**PDF disclaimer**

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.



**COPYRIGHT PROTECTED DOCUMENT**

© ISO 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
Case postale 56 • CH-1211 Geneva 20  
Tel. + 41 22 749 01 11  
Fax + 41 22 749 09 47  
E-mail [copyright@iso.org](mailto:copyright@iso.org)  
Web [www.iso.org](http://www.iso.org)

Published in Switzerland

**Contents**

Page

1 Scope.....	1
2 Normative references.....	3
3 Terms, definitions and abbreviated terms.....	5
3.1 Terms defined in ISO 1101.....	5
3.2 Terms defined in ISO 5459.....	5
3.3 Terms defined in ISO 8062-1.....	5
3.4 Terms defined in ISO 10303-1.....	5
3.5 Terms defined in ISO 10303-31.....	6
3.6 Terms defined in ISO 10303-42.....	6
3.7 Other terms and definitions.....	6
3.8 Abbreviated items.....	9
4 Information requirements.....	9
4.1 Units of functionality.....	10
4.1.1 Casting features.....	11
4.1.2 Casting prerequisites.....	13
4.1.3 Design exception.....	14
4.1.4 Die_mould_features.....	14
4.1.5 Investment_casting_features.....	15
4.1.6 Library_reference.....	16
4.1.7 Manufacturing_casting_resources.....	16
4.1.8 manufacturing_feature.....	18
4.1.9 Manufacturing_part_properties.....	18
4.1.10 Manufacturing_process_requirement_documents.....	19
4.1.11 Manufacturing_process_control_documentation.....	21
4.1.12 Measurement_limitations.....	21
4.1.13 Part_administration_data.....	23
4.1.14 Part_model.....	24
4.1.15 Process_activities.....	24
4.1.16 Process_plan.....	26
4.1.17 Quality_control.....	26
4.1.18 Requisitions.....	27
4.1.19 Shape_representation_for_casting_and_machining.....	28
4.1.20 Simulation.....	29
4.2 Application objects.....	30
4.3 Application assertions.....	295
5 Application interpreted model.....	379
5.1 Mapping specification.....	379
5.1.1 Casting features UoF.....	381
5.1.2 Casting prerequisites UoF.....	560
5.1.3 Design_exception UoF.....	606
5.1.4 Die_mould_features UoF.....	611
5.1.5 Investment_casting_features UoF.....	678
5.1.6 Library_reference UoF.....	698
5.1.7 Manufacturing_casting_resources UoF.....	706
5.1.8 Manufacturing_part_properties UoF.....	729
5.1.9 Manufacturing_process_requirement_documents UoF.....	753
5.1.10 Measurement_limitations UoF.....	777
5.1.11 Manufacturing_feature UoF.....	814

5.1.12 Manufacturing_process_control_documentation UoF .....	823
5.1.13 Part_administration_data UoF.....	842
5.1.14 Part_model UoF .....	847
5.1.15 Process_activities UoF .....	863
5.1.16 Process_plan UoF.....	914
5.1.17 Quality_control UoF .....	925
5.1.18 Requisitions UoF.....	954
5.1.19 Shape_representation_for_castings UoF.....	957
5.1.20 Simulation UoF .....	973
5.2 AIM EXPRESS short listing .....	998
6 Conformance requirements .....	1361
Annex A_(normative) AIM EXPRESS expanded listing .....	1397
Annex B (normative) AIM short names .....	1780
Annex C (normative)_Implementation method specific requirements .....	1801
Annex D (normative) Protocol Implementation Conformance Statement (proforma) .....	1802
Annex E (normative) Information object registration.....	1804
Annex F (informative) Application activity model.....	1805
Annex G (informative) Application reference model .....	1848
Annex H (informative) AIM EXPRESS-G.....	1902
Annex I (informative) Computer interpretable listings.....	1946
Annex J (informative)_Application protocol usage guide .....	1947
Bibliography .....	2237
Index .....	2239

## Figures

Figure 1 — ISO 10303 manufacturing suite of standards.....	xi
Figure 2 — Data planning model.....	xii
Figure 3 — Angular_dimension_tolerance.....	37
Figure 4 — Angular_size_dimension_tolerance.....	38
Figure 5 — Chaplet and Chaplet_pad.....	53
Figure 6 — Choke connection in a Gating_system .....	57
Figure 7 — Compound_feature .....	61
Figure 8 — Core and Core_print .....	66
Figure 9 — Curved_dimension_tolerance .....	75
Figure 10 — Diameter_dimension_tolerance.....	88
Figure 11 — Distance_along_curve_tolerance.....	98
Figure 12 — Filter connection in a Gating_system .....	124



Figure 13 — Flask .....	127
Figure 14 — Location_dimension_tolerance.....	155
Figure 15 — Pattern_plate .....	200
Figure 16 — Production_core_box with Core.....	217
Figure 17 — Radial_dimension_tolerance .....	239
Figure 18 — Riser conection to Gating_system .....	250
Figure 19 — Sprue conection to Gating_system .....	277
Figure 20 — Well and associated Gating_system .....	294
Figure F.1 — IDEF0 basic notation.....	1805
Figure F.2 — A-0 Exchange of design and manufacturing product information for cast parts.....	1827
Figure F.3 — A0 Cast a part.....	1828
Figure F.4 — A1 Sand cast a part.....	1829
Figure F.5 — A11 Design sand mould and plan process.....	1830
Figure F.6 — A111 Design sand mould .....	1831
Figure F.7 — A12 Build sand mould.....	1832
Figure F.8 — A13 Build pattern and rigging assembly .....	1833
Figure F.9 — A131 Make sand mould .....	1834
Figure F.10 — A15 Pour cool and shakeout.....	1835
Figure F.11 — A2 Die cast a part .....	1836
Figure F.12 — A12 Design die and plan die casting process .....	1837
Figure F.13 — A22 Build die .....	1838
Figure F.14 — A22 Build die .....	1839
Figure F.15 — A23 Melt, pour, cool and extract.....	1840
Figure F.16 — A3 Investment cast a part .....	1841
Figure F.17 — A31 Design investment pattern assembly and plan process.....	1842
Figure F.18 — A311 Design investment mould .....	1843
Figure F.19 — A32 Build wax or foam pattern .....	1844
Figure F.20 — A33 Invest pattern .....	1845
Figure F.21 — A34 Melt, pour, cool and Knockout.....	1846
Figure F.22 — A4 Finish and inspect.....	1847
Figure G.1 — ARM diagram (1 of 53).....	1849
Figure G.2 — ARM diagram (2 of 53).....	1850
Figure G.3 — ARM diagram (3 of 53).....	1851
Figure G.4 — ARM diagram (4 of 53).....	1852
Figure G.5 — ARM diagram (5 of 53).....	1853
Figure G.6 — ARM diagram (6 of 53).....	1854
Figure G.7 — ARM diagram (7 of 53).....	1855
Figure G.8 — ARM diagram (8 of 53).....	1856
Figure G.9 — ARM diagram (9 of 53).....	1857
Figure G.10 — ARM diagram (10 of 53).....	1858
Figure G.11 — ARM diagram (11 of 53).....	1859
Figure G.12 — ARM diagram (12 of 53).....	1860
Figure G.13 — ARM diagram (13 of 53).....	1861
Figure G.14 — ARM diagram (14 of 53).....	1862
Figure G.15 — ARM diagram (15 of 53).....	1863
Figure G.16 — ARM diagram (16 of 53).....	1864
Figure G.17 — ARM diagram (17 of 53).....	1865
Figure G.18 — ARM diagram (18 of 53).....	1866
Figure G.19 — ARM diagram (19 of 53).....	1867
Figure G.20 — ARM diagram (20 of 53).....	1868

Figure G.21 — ARM diagram (21 of 53) .....	1869
Figure G.22 — ARM diagram (22 of 53) .....	1870
Figure G.23 — ARM diagram (23 of 53) .....	1871
Figure G.24 — ARM diagram (24 of 53) .....	1872
Figure G.25 — ARM diagram (25 of 53) .....	1873
Figure G.26 — ARM diagram (26 of 53) .....	1874
Figure G.27 — ARM diagram (27 of 53) .....	1875
Figure G.28 — ARM diagram (28 of 53) .....	1876
Figure G.29 — ARM diagram (29 of 53) .....	1877
Figure G.30 — ARM diagram (30 of 53) .....	1878
Figure G.31 — ARM diagram (31 of 53) .....	1879
Figure G.32 — ARM diagram (32 of 53) .....	1880
Figure G.33 — ARM diagram (33 of 53) .....	1881
Figure G.34 — ARM diagram (34 of 53) .....	1882
Figure G.35 — ARM diagram (35 of 53) .....	1883
Figure G.36 — ARM diagram (36 of 53) .....	1884
Figure G.37 — ARM diagram (37 of 53) .....	1885
Figure G.38 — ARM diagram (38 of 53) .....	1886
Figure G.39 — ARM diagram (39 of 53) .....	1887
Figure G.40 — ARM diagram (40 of 53) .....	1888
Figure G.41 — ARM diagram (41 of 53) .....	1889
Figure G.42 — ARM diagram (42 of 53) .....	1890
Figure G.43 — ARM diagram (43 of 53) .....	1891
Figure G.44 — ARM diagram (44 of 53) .....	1892
Figure G.45 — ARM diagram (45 of 53) .....	1893
Figure G.46 — ARM diagram (46 of 53) .....	1894
Figure G.47 — ARM diagram (47 of 53) .....	1895
Figure G.48 — ARM diagram (48 of 53) .....	1896
Figure G.49 — ARM diagram (49 of 53) .....	1897
Figure G.50 — ARM diagram (50 of 53) .....	1898
Figure G.51 — ARM diagram (51 of 53) .....	1899
Figure G.52 — ARM diagram (52 of 53) .....	1900
Figure G.53 — ARM diagram (53 of 53) .....	1901
Figure H.1 — AIM EXPRESS-G diagram application_context .....	1903
Figure H.2 — AIM EXPRESS-G diagram product .....	1904
Figure H.3 — AIM EXPRESS-G diagram product_definition .....	1905
Figure H.4 — AIM EXPRESS-G diagram effectivity .....	1906
Figure H.5 — AIM EXPRESS-G diagram property_definition .....	1907
Figure H.6 — AIM EXPRESS-G diagram property_definition_representation .....	1908
Figure H.7 — AIM EXPRESS-G diagram characterized_object .....	1909
Figure H.8 — AIM EXPRESS-G diagram casting_feature_definition .....	1910
Figure H.9 — AIM EXPRESS-G diagram shape_aspect .....	1911
Figure H.10 — AIM EXPRESS-G diagram representation .....	1912
Figure H.11 — AIM EXPRESS-G diagram shape_representation .....	1913
Figure H.12 — AIM EXPRESS-G diagram representation_item .....	1914
Figure H.13 — AIM EXPRESS-G diagram point .....	1915
Figure H.14 — AIM EXPRESS-G diagram placement .....	1916
Figure H.15 — AIM EXPRESS-G diagram curve .....	1917
Figure H.16 — AIM EXPRESS-G diagram bounded_curve .....	1918
Figure H.17 — AIM EXPRESS-G diagram b-spline_curve .....	1919

Figure H.18 — AIM EXPRESS-G diagram surface.....	1920
Figure H.19 — AIM EXPRESS-G diagram bounded_surface.....	1921
Figure H.20 — AIM EXPRESS-G diagram topology.....	1922
Figure H.21 — AIM EXPRESS-G diagram shell and face_bound.....	1923
Figure H.22 — AIM EXPRESS-G diagram geometric_tolerance.....	1924
Figure H.23 — AIM EXPRESS-G diagram datum.....	1925
Figure H.24 — AIM EXPRESS-G diagram dimensional tolerance.....	1926
Figure H.25 — AIM EXPRESS-G diagram action.....	1927
Figure H.26 — AIM EXPRESS-G diagram action_method.....	1928
Figure H.27 — AIM EXPRESS-G diagram action_method_relationship.....	1929
Figure H.28 — AIM EXPRESS-G diagram action_property.....	1930
Figure H.29 — AIM EXPRESS-G diagram action_directive.....	1931
Figure H.30 — AIM EXPRESS-G diagram resource_property.....	1932
Figure H.31 — AIM EXPRESS-G diagram action_resource.....	1933
Figure H.32 — AIM EXPRESS-G diagram document.....	1934
Figure H.33 — AIM EXPRESS-G diagram group.....	1935
Figure H.34 — AIM EXPRESS-G diagram externally_defined_item.....	1936
Figure H.35 — AIM EXPRESS-G diagram approval.....	1937
Figure H.36 — AIM EXPRESS-G diagram date.....	1938
Figure H.37 — AIM EXPRESS-G diagram applied assignments.....	1939
Figure H.38 — AIM EXPRESS-G diagram address.....	1940
Figure H.39 — AIM EXPRESS-G diagram unit.....	1941
Figure H.40 — AIM EXPRESS-G diagram measure_with_unit.....	1942
Figure H.41 — AIM EXPRESS-G diagram id and description attribute.....	1943
Figure H.42 — AIM EXPRESS-G diagram name and role attribute.....	1944
Figure H.43 — AIM EXPRESS-G diagram text, label and identifier.....	1945
Figure J.1 — Manufacturing suite test case test part drawing.....	2055
Figure J.2 — Manufacturing suite test case part.....	2056

## Tables

Table 1 — Conformance class UoFs.....	1362
Table 2 — Conformance class elements.....	1375
Table B.1— AIM short names of entities.....	1780

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/ IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO 10303-223 was prepared by Technical Committee ISO/TC184, *Industrial automation systems and integration*, Subcommittee SC 4, *Industrial data*.

ISO 10303 is organized as a series of parts, each published separately. The structure of ISO 10303 is described in ISO 10303-1.

Each part of ISO 10303 is a member of one of the following series: description methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part of ISO 10303 is a member of the application protocol series.

A complete list of parts of ISO 10303 is available from the Internet:

<<http://www.tc184-sc4.org/titles/>>

## Introduction

ISO 10303 is an International Standard for the computer-interpretable representation of product information and for the exchange of product data. The objective is to provide a neutral mechanism capable of describing products throughout their life cycle. This mechanism is suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases, and as a basis for archiving.

This part of ISO 10303 is a member of the application protocol series. This part of ISO 10303 specifies an application protocol (AP) for the use of product data within a defined context that satisfies an industrial need to exchange product design, simulation, and manufacturing data for producing cast parts. In addition this part of ISO 10303 also provides for the exchange of product design and manufacturing data for value added operations to the cast parts.

This part of ISO 10303 defines the context, scope, and information requirements for the exchange of design, simulation, manufacturing features, and product information for cast parts and specifies the integrated resources necessary to satisfy these requirements. This part of ISO 10303 defines data sharing for the exchange of product specification between a customer and a metalcaster in response to a request for quote (RFQ) to supply cast parts. This part of ISO 10303 can be used to represent the product design modifications necessary for tooling design to produce the cast parts. This part of ISO 10303 provides data sharing and exchange of the original and modified product design between the metalcaster, the customer, the tool shop, and can provide data as direct input the simulation software. Recently many metalcasters use simulation software to evaluate a design and analyze a gating system to reduce the failure rate of the cast part. There is a need to reduce cost, labour time and errors in inputting the part design. This part of ISO 10303 provides a common neutral data form for this data exchange between metalcaster, customer, tool shop and simulation software. Should a customer request delivery of a machined casting to design specifications these value-added operations are supported by the machining features in this part of ISO 10303. The original part design specification from the customer and the modified part design from the metalcaster define the manufacturing features of the casting that are to be machined by a machine shop. This part of ISO 10303 interfaces directly with ISO 10303-224 and ISO 10303-219 and supports the data exchange between customer, metalcaster, and machine shop to deliver machined and inspected casting to customer specification. This part of ISO 10303 can also be used to represent the necessary process data for quality control purposes in support of the Production Part Approval Process (PPAP). Should the metalcaster require a more complete Process Plan capability, this part of ISO 10303 provides an interface to ISO 10303-240, the Process Plan for machined parts. As a member of a suite of application protocols developed to aid in the manufacture of piece parts for mechanical operations, this part of ISO 10303 has been harmonized to be integrated into that suite.

Application protocols provide the basis for developing implementations of ISO 10303 and abstract test suites for the conformance testing of AP implementations.

Clause 1 defines the scope of this part of ISO 10303 and summarizes the functionality and data covered within it. Clause 3 lists the words defined in this part of ISO 10303 and gives pointers to words defined elsewhere. An application activity model that is the basis for the definition of the scope is provided in Annex F. The information requirements of the application are specified in Clause 4 using terminology appropriate to the application. A graphical representation of the information requirements, referred to as the application reference model, is given in Annex G.

Resource constructs are interpreted to meet the information requirements. This interpretation produces the application interpreted model (AIM). This interpretation, given in 5.1, shows the correspondence between the information requirements and the AIM. The short listing of the AIM specifies the interface to the integrated resources and is given in 5.2. Note that the definitions and EXPRESS provided in the integrated resources for constructs used in the AIM can include select list items and subtypes which are not imported into the AIM. The expanded listing given in Annex A contains the complete EXPRESS for the AIM without annotation while the short names of entities is given in Annex B. Information on where to get computer interpretable EXPRESS is given in Annex I. A graphical representation of the AIM is given in Annex H. Additional requirements for specific implementation methods are given in Annex C.

Figure 2 contains the data planning model that provides a high level description of the requirements for this part of ISO 10303. This planning model was created from the in-scope data from the activities of the application activity model (AAM) and grouped into logical units of functionality. This planning model is used as a guide in developing the application reference model (ARM).

In the manufacturing product life cycle area of interest, there have been several application protocols that have been developed.

NOTE Figure 1 illustrates the application protocols that are of interest to manufacturing, and the relationship of these application protocols to each other.

As manufacturing application protocols are being developed, there has been a clear architectural plan for development of new emerging manufacturing standards. The teams developing this set of application protocols are making a concentrated effort to make sure these standards are harmonized and have a matching architecture.

The starting point for harmonization is ISO 10303-203. This part of ISO 10303 first developed Boundary Representation (B-Rep) geometry. This B-Rep geometry resulted in the development of Application Integrated Construct (AIC) for Advanced Brep – ISO 10303-514. This geometry is used throughout the manufacturing application protocols.

The next point of harmonization is ISO 10303-224. This part of ISO 10303 first developed machining features (ISO 10303-522), geometric tolerances (ISO 10303-519), dimensional tolerances, definitions for implicit shape definitions, and property definitions for materials, process, part, and surface finish.

Since the development of ISO 10303-224, several application protocols have been developed to use constructs defined by ISO 10303-224. There has been complete harmonization of constructs with all of the other manufacturing application protocols.

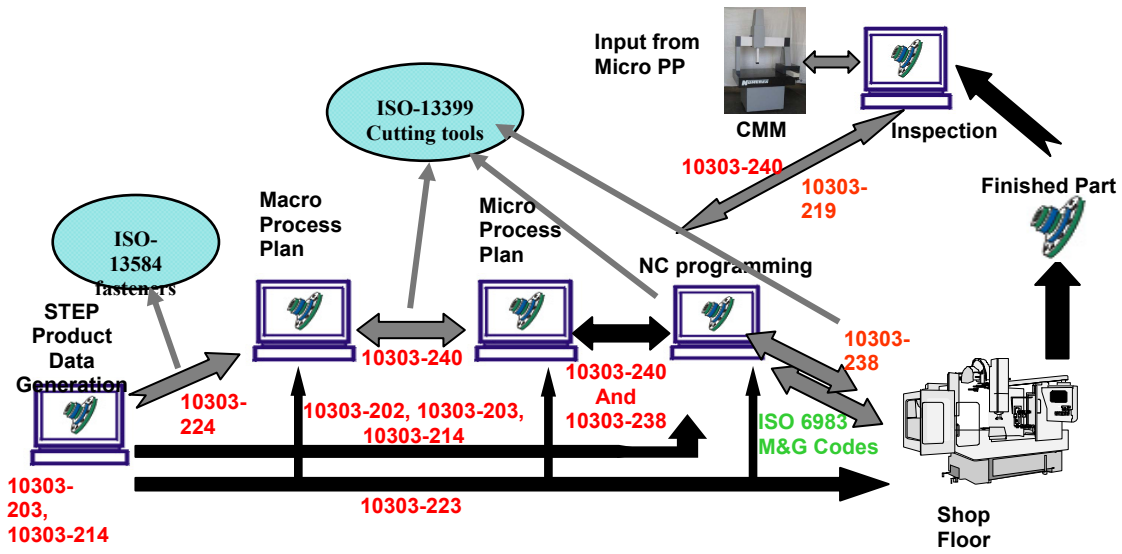
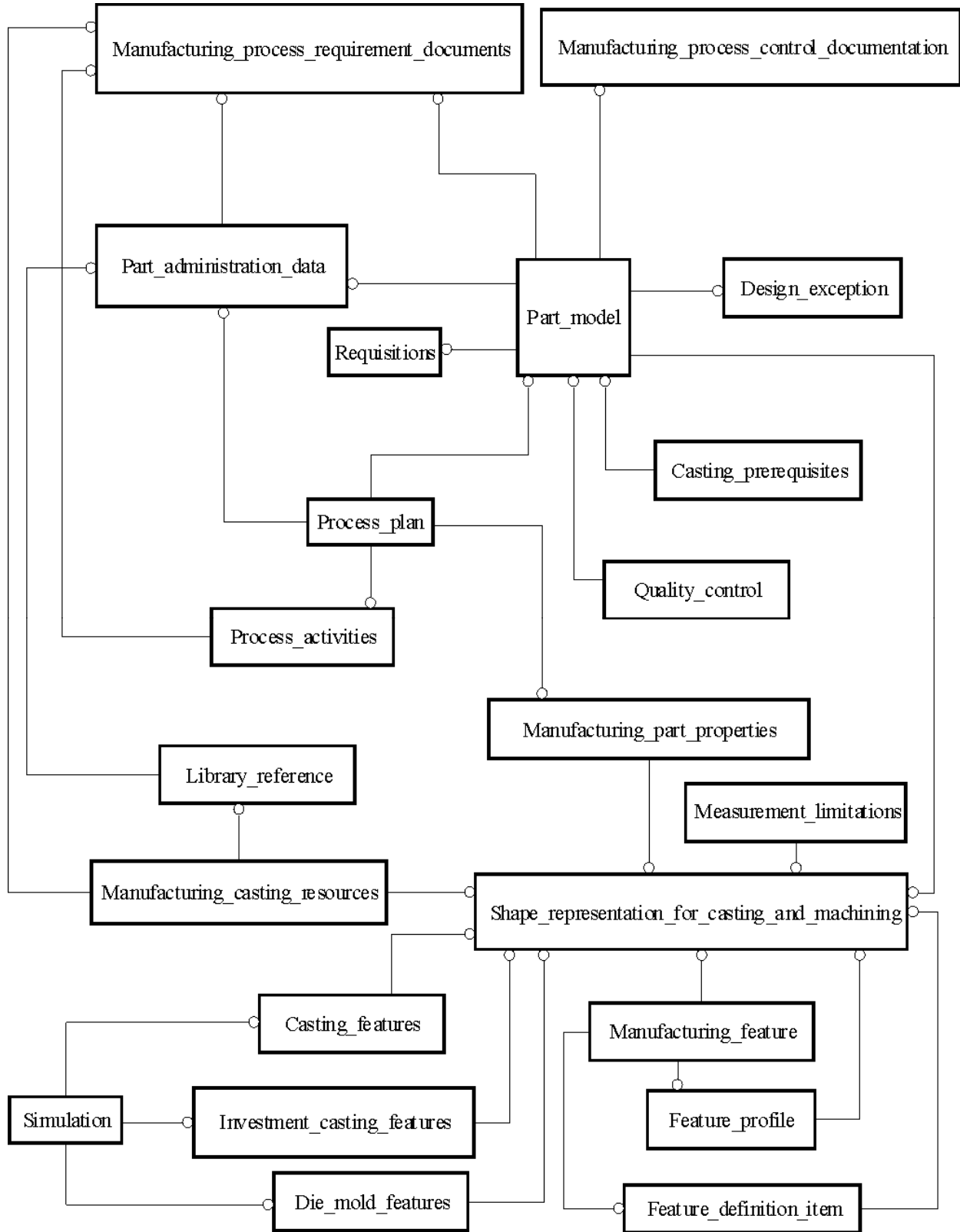


Figure 1 — ISO 10303 manufacturing suite of standards



**Figure 2 — Data planning model**



---

# **Industrial automation systems and integration — Product data representation and exchange — Part 223: Application protocol: Exchange of design and manufacturing product information for cast parts**

## **1 Scope**

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for exchange, archiving and sharing of design and manufacturing product information for cast parts.

NOTE 1 The application activity model, in Annex F, provides a graphical representation of the processes and information flows which are the basis for the definition of the scope of this part of ISO 10303.

The following are within the scope of this part of ISO 10303:

- the manufacturing of a cast part made by sand, die, and investment casting processes;
- parts that are to be manufactured by casting processes;
- design data for cast parts, including geometry, materials, tolerances, required physical and mechanical properties, required tests;
- casting features for defining shapes necessary for casting processes;

NOTE 2 The casting feature set is defined in this part of ISO 10303.

- manufacturing features for defining shapes necessary for machining processes;
- manufacturing features for defining shapes necessary for casting process;
- explicit representation of the 3D shape of casting features through bounded geometry representations;
- geometric and dimensional tolerances of the parts being manufactured;
- materials, and properties of the parts being manufactured;
- characterization of products used to make cast parts, including moulds, dies, equipment, materials, and consumable items;

ISO 10303-223:2008 (E)

- customer order administrative data to track receipt of an order for a cast part to the shop floor, but not including tracking of the order on the shop floor;
- approval data to authorize the manufacture of a cast part;
- requisition administrative data to identify requirements and track the status of materials and equipment needed to manufacture a cast part;
- work order data to track and identify the status of a cast part;
- tracking the state of raw stock for documenting the manufacturing history of a cast part;
- tracking a design exception notice of a cast part;

NOTE 3 The design exception notice relates to discrepancies in the features used to describe a cast part's shape.

- process plans for parts that are made by sand, die, and investment casting processes:
  - process data for part routing which includes manufacturing process and setup sequencing;
  - process data for operation;
- work instructions for the tasks required to manufacture a cast part, which include:
  - references to the resources required to perform the work;
  - the sequences of the work instructions;
  - relationships of the work to the part geometry;
- specifications for patterns and die assemblies;
- input to and output from casting process simulation software;
- data exchange between customer and foundry, within the foundry, and between foundry and supplier;
- use of data for foundry automation and shop floor control;
- use of data for archiving of design and manufacturing data for cast parts.

NOTE 4 Data supported by this part of ISO 10303 might need to be archived to meet legal and regulatory requirements, and to meet quality objectives.

The following are outside the scope of this part of ISO 10303:

- centrifugal cast parts;
- data describing rules, guidelines and expert knowledge used to design and manufacture cast parts;
- data describing why a particular design or manufacturing decision was made;
- forging data;
- pit moulding;
- shop floor scheduling data;
- process plans for making patterns, dies, and other tooling;
- algorithms used to obtain simulation results.

## 2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 286-1, *ISO system of limits and fits — Part 1: Bases of tolerances, deviations, and fits*.

ISO 286-2, *ISO system of limits and fits — Part 2: Tables of standard tolerance grades and limit deviations for holes and shafts*.

ISO 1101, *Geometrical Product Specifications (GPS) — Geometrical tolerancing — Tolerances of form, orientation, location and run-out*.

ISO 5459, *Geometrical product specifications (GPS) — Geometrical tolerancing — Datums and datum-systems*.

ISO/IEC 8824-1, *Information technology — Abstract Syntax Notation One (ASN.1) — Part 1: Specification of basic notation*

ISO 8062-1:2007, *Geometrical product specifications (GPS) — Dimensional and geometrical tolerances for moulded parts — Part 1: Vocabulary*.

ISO 10303-1, *Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles*.

ISO 10303-11:2004, *Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual*.

ISO 10303-223:2008 (E)

ISO 10303-21:2002, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure.*

ISO 10303-31, *Industrial automation systems and integration — Product data representation and exchange — Part 31: Conformance testing methodology and framework: General concepts.*

ISO 10303-41, *Industrial automation systems and integration — Product data representation and exchange — Part 41: Integrated generic resource: Fundamentals of product description and support.*

ISO 10303-42, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation.*

ISO 10303-43, *Industrial automation systems and integration — Product data representation and exchange — Part 43: Integrated generic resource: Representation structures.*

ISO 10303-45, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resource: Material and other engineering properties.*

ISO 10303-47, *Industrial automation systems and integration — Product data representation and exchange — Part 47: Integrated generic resource: Shape variation tolerances.*

ISO 10303-49, *Industrial automation systems and integration — Product data representation and exchange — Part 49: Integrated generic resources: Process structure and properties.*

ISO 10303-224:2006, *Industrial automation systems and integration — Product data representation and exchange — Part 224: Application protocol: Mechanical product definition for process planning using machining features.*

ISO 10303-238, *Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Application interpreted model for computerized numerical controllers.*

ISO 10303-240:2005, *Industrial automation systems and integration — Product data representation and exchange — Part 240: Application protocol: Process plans for machined products.*

ISO 10303-511, *Industrial automation systems and integration — Product data representation and exchange — Part 511: Application interpreted construct: Topologically bounded surface.*

ISO 10303-514, *Industrial automation systems and integration — Product data representation and exchange — Part 514: Application interpreted construct: Advanced boundary representation.*

ISO 10303-519, *Industrial automation systems and integration — Product data representation and exchange — Part 519: Application interpreted construct: Geometric tolerances.*

ISO 10303-522, *Industrial automation systems and integration — Product data representation and exchange — Part 522: Application interpreted construct: Machining features.*

### **3 Terms, definitions and abbreviated terms**

#### **3.1 Terms defined in ISO 1101**

For the purposes of this document, the following terms defined in ISO 1101 apply.

- dimension;
- tolerance.

#### **3.2 Terms defined in ISO 5459**

For the purposes of this document, the following terms defined in ISO 5459 apply.

- datum.

#### **3.3 Terms defined in ISO 8062-1**

For the purposes of this document, the following terms defined in ISO 8062-1 apply.

- mismatch.

#### **3.4 Terms defined in ISO 10303-1**

For the purposes of this document, the following terms defined in ISO 10303-1 apply.

- application;
- application activity model (AAM);
- application interpreted model (AIM);
- application object;
- application protocol (AP);
- application reference model (ARM);
- implementation method;
- integrated resource;
- model;
- product;

ISO 10303-223:2008 (E)

- product data;
- protocol implementation conformance statement (PICS);
- unit of functionality (UoF).

### **3.5 Terms defined in ISO 10303-31**

For the purposes of this document, the following terms defined in ISO 10303-31 apply.

- conformance class;
- conformance testing.

### **3.6 Terms defined in ISO 10303-42**

For the purposes of this document, the following terms defined in ISO 10303-42 apply.

- boundary representation (Brep) solid model;
- manifold solid boundary representation.

### **3.7 Other terms and definitions**

For the purposes of this document, the following terms and definitions apply.

#### **3.7.1**

##### **casting feature**

shape that represents the definition of a pattern and casting item necessary to create a cast part

#### **3.7.2**

##### **component**

product identified from a group of associated products

#### **3.7.3**

##### **core box**

structure that has the shape of the desired core to be produced, the female shape of the desired core (male)

#### **3.7.4**

##### **core cavity**

interior form of a core box that gives shape to a core

#### **3.7.5**

##### **core print**

part of a pattern or core box that has been so designed as to form a seat to locate and support a core within the mould

**3.7.6****design phase**

period of time when the engineering representation of a product is dynamic

**3.7.7****die casting**

castings process of injecting molten metal into a die, the metal being held under pressure until it solidifies into a net shape metal part

NOTE This category includes high and low pressure die casting processes. The high pressure processes include normal die casting, squeeze casting, semi-solid casting and vacuum die casting; and usually use large, complex moulds constructed of heat treated tool steel or other high strength materials. The low pressure processes use a variety of methods to introduce metal into the mould, including gravity pour, vacuum and pressure lift, and tilt pour processes. Because high pressures are not employed, special cast iron alloys are usually used to construct the moulds.

**3.7.8****foundry representative**

sales representative that is under contract with a foundry to contact companies that would require cast parts

**3.7.9****green sand**

sand that is naturally bonded, or compounded moulding sand mixture that has been tempered with water for use while still in the damp or wet condition and contains sufficient refractory clay substance to bond strongly without destroying the venting quality when rammed to the required degree of hardness

**3.7.10****investment casting**

metal casted into a mould produced by surrounding, or investing, an expendable pattern with a refractory slurry coating that sets at room temperature, the wax or plastic pattern then being removed through the use of heat prior to filling the mould with liquid metal

NOTE This category includes the lost wax investment process, where a consumable wax pattern is coated or invested with a plaster or ceramic shell mould. Also, plastics and foams have been used to produce the pattern. This latter process is also included in this part of ISO 10303 and is called lost foam, expendable pattern or evaporative pattern casting.

**3.7.11****manufacturing feature**

shape which represents volumes of material that are removed from a part by machining or that result from machining

**3.7.12****mechanical part**

physical object that can be formed from material into a static shape

**3.7.13**

**modified part design**

revised design of a part design to permit a part to be cast

EXAMPLE Changing the basic shape of a part by adding a slope to all vertical surfaces, so the cast part can be removed from a mould or die.

**3.7.14**

**mould**

cavity where molten metal is poured to produce a cast part

**3.7.15**

**pattern**

wood, metal, plaster, foam, plastic or wax shape used to form a mould cavity

**3.7.16**

**refractory material**

ability of the casting material to withstand high temperatures without fusing or breaking down

**3.7.17**

**sand casting**

casting process that uses a pattern, also called a match plate or pressure plate, to make a disposable mould for each casting poured

NOTE This category includes and covers all unbounded green sand processes, bonded sand processes and plaster mould processes. The term "sand" used in this part of ISO 10303 includes green sand, bonded sand, plaster and other similar moulding materials used to create a mould. This category also includes semi-permanent moulding processes, in the sense that the permanent part of the mould is treated as a specially designed chill. Also included is the sand core package assembly casting process, where a number of bonded sand cores are assembled to form a continuous cavity. In this case no pattern is employed. The pattern is in effect divided up among the different cores.

**3.7.18**

**solid model**

three-dimensional object with a well defined inside and outside separated by a two-dimensional boundary

**3.7.19**

**tooling design**

configuration of one or more devices that will produce a cast part

**3.7.20**

**value-added operation**

process or procedure that is performed on a cast part after foundry operations have been completed

EXAMPLE Machining of the casting, heat treatment of the casting, painting the casting.



### 3.8 Abbreviated items

For the purposes of this document, the following abbreviated terms apply:

AAM	application activity model
AIC	application interpreted construct
AIM	application interpreted model
AP	application protocol
ARM	application reference model
Brep	boundary representation
EPS	Expandable Polystyrene foam
ICOM	input, control, output, mechanism
NDE	non-destructive evaluations
NDT	non-destructive testing
Ngon	N number of sides polygon
PICS	protocol implementation conformance statement
PPAP	production part approval process
RFQ	request for quote
UoF	unit of functionality

## 4 Information requirements

This clause specifies the information required for the definition of product data for Exchange of design and manufacturing product information in the context of cast parts.

ISO 10303-223:2008 (E)

The information requirements are specified as a set of units of functionality, application objects, and application assertions. These assertions pertain to individual application objects and to relationships between application objects. The information requirements are defined using the terminology of the subject area of this part of ISO 10303.

NOTE 1 A graphical representation of the information requirements is given in Annex G.

NOTE 2 The information requirements correspond to those of the activities identified as being in the scope of the part of ISO 10303 in Annex F.

NOTE 3 The mapping specification specified in 5.1 shows how the information requirements are met using the integrated resources and application interpreted constructs of this International Standard. The use of the integrated resources and application interpreted constructs introduces additional requirements that are common to application protocols.

## 4.1 Units of functionality

This subclause specifies the units of functionality for the Exchange of design and manufacturing product information for cast parts application protocol.

This part of ISO 10303 specifies the following units of functionality:

- casting\_features;
- casting\_prerequisites;
- design\_exception;
- die\_mould\_features;
- investment\_casting\_features;
- library\_reference;
- manufacturing\_casting\_resources;
- manufacturing\_feature;
- manufacturing\_part\_properties;
- manufacturing\_process\_requirement\_documents;
- manufacturing\_process\_control\_documentation;
- measurement\_limitations;
- part\_administration\_data;
- part\_model;

- process\_activities;
- process plan;
- quality\_control;
- requisitions;
- shape\_representation\_for\_casting\_and\_machining;
- simulation.

### 4.1.1 Casting features

The casting\_feature UoF contains the information necessary to identify shapes that represent the definition of patterns and casting items necessary to create a cast part.

The following application objects are used by the casting\_feature UoF:

- Casting\_design\_feature;
- Casting\_insert;
- Chaplet;
- Chaplet\_pad;
- Chill;
- Choke;
- Connection\_transition;
- Core;
- Core\_box\_tooling;
- Core\_master;
- Core\_print;
- Ejector\_box;
- Ejector\_design\_feature;
- Ejector\_pins;

ISO 10303-223:2008 (E)

- Ejector\_plate;
- Ejector\_system;
- Feeder;
- Filter;
- Fixturing\_tab;
- Flask;
- Flaskless;
- Gating\_design\_feature;
- Gating\_system;
- General\_rib;
- Ingate\_contacts;
- Ingate;
- Item\_size;
- Loose\_piece;
- Master\_pattern;
- Mould\_box;
- Mould\_locks;
- Pattern\_plate;
- Pin\_center;
- Pin\_tips;
- Planar\_rib;
- Production\_core\_box;
- Production\_pattern\_definition;
- Reset\_or\_return\_pins;

- Rib;
- Riser\_contact;
- Risers;
- Runner;
- Sand\_cast\_design\_feature;
- Sand\_casting\_tooling;
- Sand\_mould;
- Sprue;
- Stop\_pin;
- Vent;
- Well.

#### 4.1.2 Casting\_prerequisites

The casting\_prerequisites UoF specifies the necessary criteria that are to be satisfied for the production of a cast part.

NOTE most the conditions are stated either by the customer and the machine shop except for the Process\_requirement which is stated by the foundry.

The following application objects are used by the casting\_prerequisites UoF

- Casting\_service\_data;
- Corrosion\_service;
- Customer\_casting\_requirement;
- Elevated\_temperature\_service;
- Finishing\_or\_machining\_operations;
- Heat\_treat\_requirement;
- Inspection\_or\_test\_requirement;
- Inspection\_plan;

ISO 10303-223:2008 (E)

- Machining\_allowance;
- Mechanical\_property\_requirements;
- Mechanical\_stress;
- Mismatch\_tolerance;
- Pattern\_equipment\_available;
- Payment\_and\_shipping;
- Process\_requirement;
- Reporting\_requirement;
- Special\_inspection\_requirement;
- Tolerance\_requirement.

### **4.1.3 Design\_exception**

The design\_exception UoF contains the application objects used for documentation required for issuing an error report for a problem that was discovered in the creation of a process plan, and a solution to the problem as it pertains to the regeneration of input data for further cast parts.

NOTE This unit of functionality is completely common with ISO 10303-224:2006. All objects are completely harmonized with 10303-224:2006.

The following application objects are used by the design\_exception UoF:

- Design\_exception\_notice;
- Engineering\_change\_order;
- Engineering\_change\_proposal;
- Request\_for\_clarification.

### **4.1.4 Die\_mould\_features**

The die\_mould\_feature UoF contains the information necessary to identify explicit shapes and implicit attributes required to define die moulds and features for a die mould.

The following application objects are used by the die\_mould\_feature UoF:

- Cooling\_port;
- Die\_design\_feature;
- Die\_cast\_master;
- Die\_cast\_tooling;
- Die\_clamping;
- Die\_core;
- Die\_master;
- Die\_mould\_tooling;
- Die\_mould\_vent;
- Production\_die\_cast\_mould;
- Production\_die\_mould;
- Production\_tool;
- Secondary\_tooling;
- Shot\_sleeve;
- Slide.

#### **4.1.5 Investment\_casting\_features**

The investment\_casting\_features UoF contains the information necessary to identify explicit shapes and implicit attributes required to define investment casting and features for an investment\_casting.

The following application objects are used by the investment\_casting\_features UoF:

- Iinvestment\_casting\_master;
- Investment\_casting\_tooling;
- Investment\_casting\_tooling\_definition;
- Investment\_design\_feature;

ISO 10303-223:2008 (E)

- Investment\_mould;
- Lost\_foam\_casting\_die;
- Production\_investment\_cast\_mould;
- Sprue\_and\_runner\_mould.

#### **4.1.6 Library\_reference**

The library\_reference UoF provides the capability and mechanisms by which references can be made to information in external libraries.

NOTE 1 A supplier library of part information is a type of library that may be referenced.

NOTE 2 This unit of functionality is completely common with ISO 10303-224:2006. All objects are completely harmonized with ISO 10303-224:2006.

The following application objects are used by the library\_reference UoF:

- BSU;
- Class\_BSU;
- Externally\_defined\_representation;
- Library\_part\_assignment;
- Property\_BSU;
- Property\_value;
- Supplier\_BSU.

#### **4.1.7 Manufacturing\_casting\_resources**

The manufacturing\_casting\_resources UoF contains the definitions of tools, fixtures, and jigs used to accomplish the casting and fabrication of a part.

NOTE This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240. These objects are: Controller, Resource\_with\_material and Resource\_with\_representation. The object Generic\_casting\_resource is common with Generic\_manufacturing\_resource found in ISO 10303-240:2005, only the name is different.



The following application objects are used by the manufacturing\_castingl\_resources UoF:

- Bankout\_frame;
- Controller;
- Core\_bench;
- Core\_box\_vent;
- Core\_conveyer\_system;
- Core\_equipment;
- Core\_machine;
- Core\_setter;
- Dryer;
- Ejector\_equipment;
- Equipment;
- Equipment\_size;
- Finishing;
- Generic\_casting\_resource;
- In\_facility\_location;
- Machine\_mounting;
- Melting\_equipment;
- Moulding\_equipment;
- Resource\_with\_material;
- Resource\_with\_representation;
- Shakeout.

### 4.1.8 manufacturing\_feature

The manufacturing\_feature UoF contains the information necessary to identify shapes which represent volumes of material that shall be removed from a part by machining or shall result from machining.

The following application objects are used by the manufacturing\_feature UoF:

- Compound\_feature;
- Compound\_feature\_element;
- Compound\_feature\_relationship;
- Feature;
- Manufacturing\_design\_feature;
- Machining\_feature;
- Manufacturing\_feature;
- Manufacturing\_feature\_group;
- Transition\_feature.

### 4.1.9 Manufacturing\_part\_properties

The manufacturing\_part\_properties UoF contains the description of characteristics of the part that is being defined. These characteristics specify requirements for manufacturing that apply to either the state of the part at a particular time prior to or after the manufacture of the part, or a process that is required to be executed during the manufacture of the part.

NOTE This unit of functionality is partially common with ISO 10303-224:2006. The objects in common are completely harmonized with ISO 10303-224:2006. These objects are: Descriptive\_parameter, Hardness, Material, Material\_property, Numeric\_parameter, Numeric\_parameter\_with\_tolerance, Part\_property, Process\_property, Property, Property\_parameter and Surface\_property.

The following application objects are used by the manufacturing\_part\_properties UoF:

- Casting\_weight\_requirement;
- Composition\_element;
- Data\_curve;
- Descriptive\_parameter;
- Geometry\_characteristic;

- Hardness;
- Material;
- Material\_composition;
- Material\_property;
- Material\_structure;
- Material\_structure\_element;
- Numeric\_parameter;
- Numeric\_parameter\_with\_tolerance;
- Numeric\_range;
- Part\_property;
- Process\_property;
- Property;
- Property\_relationship;
- Property\_parameter;
- Surface\_property;

#### **4.1.10 Manufacturing\_process\_requirement\_documents**

The manufacturing\_process\_requirement\_documents UoF contains information that identifies product data for control documents. These documents include contracts, drawings, additional data for process plans, and other supporting documentation for process plans. These documents may be in either digital or hardcopy format.

NOTE 1 This unit of functionality is partially common with ISO 10303-224:2006. The objects in common are completely harmonized with ISO 10303-224:2006. These objects are: Part\_dimensioning\_standard, Specification and Specification\_usage\_constraint.

NOTE 2 This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240. These objects are: Digital\_file, Document\_assignment, Document\_file, Document\_file\_properties, Document\_file\_relationship, Design\_reference, Executable, External\_file\_identification, External\_schema\_definition, Hardcop, Illustration, Partial\_document\_assignment, Part\_dimensioning\_standard, Special\_instruction, Specification, Supplemental\_documentand View\_reference.

ISO 10303-223:2008 (E)

The following application objects are used by the manufacturing\_process\_requirement\_documents UoF:

- Customer\_design\_deficiency\_report;
- Customer\_tool\_design\_report;
- Design\_reference;
- Digital\_file;
- Document\_assignment;
- Document\_file;
- Document\_file\_properties;
- Document\_file\_relationship;
- Executable;
- External\_file\_identification;
- External\_schema\_definition;
- Hardcopy;
- Illustration;
- Part\_dimensioning\_standard;
- Product\_quality\_report;
- Proof\_report;
- Quality\_acceptance\_report;
- Simulation\_exception\_report;
- Simulation\_report;
- Simulation\_tooling\_report;
- Special\_instruction;
- Specification;
- Specification\_usage\_constraint;

- Supplemental\_document;
- View\_reference.

#### **4.1.11 Manufacturing\_process\_control\_documentation**

The manufacturing\_process\_control\_documentation UoF contains information that identifies product data order information. These documents include work order information from an internal or external customer and internal manufacturing operations documentation that allows for the tracking of the work on a part's manufacture.

NOTE This unit of functionality is completely common with ISO 10303-224:2006. All objects are completely harmonized with ISO 10303-224:2006.

The following application objects are used by the manufacturing\_process\_control\_documentation UoF:

- Customer\_order;
- Digital\_technical\_data\_package\_work\_order;
- Ordered\_part;
- Pedigree\_creation\_order;
- Project\_order;
- Request\_for\_quotation
- Resource\_acquisition\_order;
- Shop\_work\_order.

#### **4.1.12 Measurement\_limitations**

The measurement\_limitations UoF contains the information necessary to identify the important sizes of the measured relationships between aspects of a part's shape or between an aspect of a part's shape and a reference shape that does not comprise the shape of the part, and the acceptable deviation from that size or relationship for the purpose of manufacturing.

NOTE This unit of functionality is completely common with ISO 10303-224:2006. All objects are completely harmonized with ISO 10303-224:2006.

The following application objects are used by the measurement\_limitations UoF:

- Angular\_dimension\_tolerance;
- Angular\_size\_dimension\_tolerance;

ISO 10303-223:2008 (E)

- Common\_datum;
- Curved\_dimension\_tolerance;
- Datum;
- Datum\_feature;
- Datum\_target;
- Datum\_target\_set;
- Diameter\_dimension\_tolerance;
- Dimensional\_tolerance;
- Distance\_along\_curve\_tolerance;
- Externally\_defined\_size\_dimension;
- Geometric\_tolerance;
- Geometric\_tolerance\_precedence\_relationship;
- Height\_dimension;
- Length\_dimension;
- Limits\_and\_fits;
- Location\_dimension\_tolerance;
- Location\_tolerance;
- Material\_condition\_modifier;
- Placed\_target;
- Plus\_minus\_value;
- Projection;
- Radial\_dimension\_tolerance;
- Size\_tolerance;
- Target\_area;

- Target\_circle;
- Target\_line;
- Target\_point;
- Target\_rectangle;
- Thickness\_tolerance;
- Tolerance\_limit;
- Tolerance\_range;
- Tolerance\_value;
- Tolerance\_zone;
- Tolerance\_zone\_definition;
- Width\_dimension.

#### 4.1.13 Part\_administration\_data

The part\_administration\_data UoF contains the information used in the management of product data.

NOTE 1 This unit of functionality is partially common with ISO 10303-224:2006. The objects in common are completely harmonized with ISO 10303-224:2006. These objects are: Approval, Organization, Person and Person\_in\_organization.

NOTE 2 This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240. These objects are: Status\_authority.

The following application objects are used by the part\_administration\_data UoF:

- Address;
- Approval;
- Date;
- Organization;
- Person;
- Person\_in\_organization;
- Status\_authority.

#### 4.1.14 Part\_model

The part\_model UoF contains the information necessary to identify the part that is to be input to the cast parts function and identify the association of properties with that part. Additionally, information pertaining to feedback about the quality or necessary revisions to the product data is represented by this UoF.

The following application object is used by the part\_model UoF:

- Cast\_part;
- Cast\_part\_with\_rigging;
- Design\_part;
- Master\_sample;
- Part\_relationship;
- Part\_version;
- Red\_line\_part;
- Tooling.

#### 4.1.15 Process\_activities

The process\_activities UoF contains the information which defines a specific manufacturing action. Manufacturing actions deal with the transformation and inspection that takes place in the sequential process of machining a source material into a final part. These manufacturing actions include processes, operations and tasks associated with material removal by numerically controlled machine tools.

NOTE This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240. These objects are: Activity, Allowed\_time, Alternate\_activity, Ancillary\_activity, Ancillary\_setup, Continuous\_process, Manufacturing\_process, Performance\_rate, Process\_activity and Production\_rate.

The following application objects are used by the process\_activities UoF:

- Activity;
- Allowed\_time;
- Alternate\_activity;
- Ancillary\_activity;
- Ancillary\_setup;



- Assembly;
- Bench\_inspection;
- Casting\_activity;
- General\_foundry\_process;
- Customer\_simulation;
- Dimensional\_measurement\_inspection;
- Equipment\_process;
- Equipment\_setup;
- Fabrication;
- Heat\_treat;
- Impregnation\_activity;
- Inspection\_activity;
- Machining\_activity;
- Manufacturing\_process;
- Metalcaster\_simulation;
- Non\_casting\_operations;
- Non\_equipment\_process;
- Non\_permanent\_moulding\_process;
- Permanent\_moulding\_process;
- Performance\_rate;
- Production\_rate;
- Production\_welding;
- Setup\_activity;
- Simulation\_process;

ISO 10303-223:2008 (E)

- Surface\_finish;
- Tooling\_process;

#### **4.1.16 Process\_plan**

The process\_plan UoF contains the application objects used for defining a process plan, revisions of the process plan, machining feature classification structure and manufacturing process and setup sequencing.

NOTE This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240. These objects are: Alternate\_process\_plan, Continuous\_process, Process\_plan, Process\_plan\_security, Process\_plan\_version, Revision.

The following application objects are used by the process\_plan UoF:

- Alternate\_process\_plan;
- Continuous\_process;
- Process\_plan;
- Process\_plan\_security;
- Process\_plan\_version;
- Revision.

#### **4.1.17 Quality\_control**

The quality\_control UoF specifies the inspection and test results obtained from a sample set of lots or heats of castings.

The following application objects are used by the quality\_control UoF:

- Activity\_execution\_result;
- Activity\_result\_record;
- Casting\_per\_order;
- Casting\_verification;
- Checking\_aid\_tool;
- Composition\_element\_record;
- Composition\_inspection;

- Core\_assembly;
- Equipment\_setting\_record;
- Equipment\_usage;
- First\_article;
- Heat;
- Inspection\_or\_test\_result;
- Lot;
- Material\_usage;
- Material\_usage\_record;
- Process\_parameter\_record;
- Property\_inspection;
- Sampled\_set;
- Shape\_dimensions;
- Other\_inspection;
- Tolerance\_inspection;
- Tool\_verification;

#### **4.1.18 Requisitions**

The requisitions UoF contains the information necessary to identify documents generated by a manufacturing organization that specify an order for obtaining necessary manufacturing resources.

NOTE This unit of functionality is partially common with ISO 10303-224:2006. The objects in common are completely harmonized with ISO 10303-224:2006. These objects are: Indirect\_stock\_requisition, Material\_requisition, and Requisition.

The following application objects are used by the requisitions UoF:

- Die\_mould\_requisition;
- Indirect\_stock\_requisition;
- Material\_requisition;

ISO 10303-223:2008 (E)

- Pattern\_tooling\_requisition;
- Requisition.

#### **4.1.19 Shape\_representation\_for\_casting\_and\_machining**

The shape\_representation\_for\_casting\_and\_machining UoF contains the physical definition of initial and final form of the part. This definition is given via a parametric method for features, and additional geometric and topological definitions.

NOTE This unit of functionality is partially common with ISO 10303-240:2005. The objects in common are completely harmonized with ISO 10303-240:2005. These objects are: Direction\_element, Face\_shape\_element, Face\_shape\_element\_relationship, Geometric\_model, Location\_element, Model\_element, Object\_element\_shape\_representation, Orientation, Part\_shape, Path\_element, Planar\_element, Shape\_aspect, Shape\_aspect\_representation, Shape\_element and Shape\_representation\_type.

The following application objects are used by the shape\_representation\_for\_casting\_and\_machining UoF:

- Base\_shape;
- Casting\_envelope\_requirement;
- Curve\_point;
- Direction\_element;
- Drafted\_surface;
- Edge\_shape\_element;
- Face\_shape\_element;
- Face\_shape\_element\_relationship;
- Geometric\_model;
- Location\_element;
- Model\_element;
- Object\_element\_shape\_representation;
- Orientation;
- Part\_shape;
- Parting\_surface;
- Path\_element;

- Planar\_element;
- Shape\_aspect;
- Shape\_aspect\_representation;
- Shape\_element;
- Shape\_representation\_type.

#### **4.1.20 Simulation**

The simulation UoF specifies the input data and the output data for a casting simulation model.

The following application objects are used by the simulation UoF:

- Boundary\_condition;
- Condition\_or\_assumption;
- Defect\_prediction;
- Integration\_interval;
- Meshing\_condition;
- Simulated\_property;
- Simulation\_input;
- Simulation\_input\_region;
- Simulation\_output;
- Simulation\_output\_region;
- Simulation\_result;
- Simulation\_run;
- Simulation\_run\_relationship;

ISO 10303-223:2008 (E)

- Simulation\_software;
- Simulation\_unit;
- Simulation\_unit\_state;

## 4.2 Application objects

This subclause specifies the application objects for the Exchange of design and manufacturing product information for cast parts application protocol. Each application object is an atomic element that embodies a unique application concept and contains attributes specifying the data elements of the object. The application objects and their definitions are given below.

### 4.2.1 Activity

An Activity is a discrete task, or set of tasks, that is executed in the manufacturing of a product. Each Activity is either an Casting\_activity (see 4.2.21), Activity\_setup (see 4.2.280), Alternate\_activity (see 4.2.6) or Machining\_activity (see 4.2.174).

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.1.

The data associated with an Activity are the following:

- activity\_number;
- activity\_information;
- consumable\_resource;
- description;
- duration;
- execution\_results;
- frequency;
- graphics\_representation;
- name;
- output\_item;
- performed\_by\_organization\_id;
- process\_specification.

#### **4.2.1.1 activity\_information**

The `activity_information` specifies the `Special_instruction` (see 4.2.307) for the purpose of planning a machining activity. The `activity_information` need not be specified for a particular `Special_instruction`. here may be more than one `activity_information` for a `Special_instruction`. See 4.3.9 for the application assertion.

#### **4.2.1.2 activity\_number**

The `activity_number` specifies the unique identifier for an operation, task, or step, within an `process_plan_version`.

#### **4.2.1.3 consumable\_resource**

The `consumable_resources` specifies the `Generic_casting_resource` (see 4.2.136) that defines the additional resources that may be necessary to perform the activity. The `consumable_resource` need not be specified for a particular Activity. See 4.3.5 for the application assertion.

#### **4.2.1.4 description**

The description specifies the word, or group of words, that relate detail information about the Activity.

#### **4.2.1.5 duration**

The duration specifies the `Performance_rate` (see 4.2.225) which is the amount of time necessary to complete the task. See 4.3.8 for the application assertion.

#### **4.2.1.6 execution\_results**

The `execution_results` specifies the outcome of the completed Activity. See 4.3.1 for the application assertion.

#### **4.2.1.7 frequency**

The frequency specifies the number of repetitions a process plan is performed for a particular part. The frequency need not be specified for a particular Activity.

#### **4.2.1.8 graphics\_representation**

The `graphics_representation` specifies the visual representation of information necessary for the activity. There may be more than one `graphics_representation` for a `Design_reference` (see 4.2.77), `Illustration` (see 4.2.147), or `View_reference` (see 4.2.337). The `graphics_representation` need not be specified for a particular Activity. See 4.3.4, 4.3.6, and 4.3.11 for the application assertion.

#### **4.2.1.9 name**

The name specifies the word, or group of words that make up the unique designation of an Activity.

#### **4.2.1.10 output\_item**

The output item specifies the type of casting produced by the Activity. See 4.3.2 and 4.3.3 for the application assertion.

#### **4.2.1.11 performed\_by\_organization\_id**

The performed\_by\_organization\_id specifies the organization responsible for performing the Activity. The performed\_by\_organization\_id need not be specified for a particular Activity. See 4.3.7 for the application assertion.

#### **4.2.1.12 process\_specification**

The process\_specification specifies the specification or standard describing specific information about the Activity. The process\_specification need not be specified for a particular Activity. See 4.3.10 for the application assertion.

EXAMPLE The schedule of heat treatments established for aluminum-based alloys by the Aluminum Association (United States).

### **4.2.2 Activity\_execution\_result**

An Activity\_execution\_result is the performance of a Process\_plan (see 4.2.236) to manufacture a cast part.

NOTE The Activity\_execution\_result may be used to record differences between planned and actual conditions, as well as specific conditions existed, when a range of conditions are allowed by the process plan.

The data associated with an Activity\_execution\_result are the following:

- activity\_results;
- description;
- part\_measured.

#### **4.2.2.1 activity\_results**

The activity\_results specifies the records for how a manufacturing operation was carried out. See 4.3.12 for the application assertion.



### 4.2.2.2 description

The description specifies the written account of the `Activity_execution_result`.

### 4.2.2.3 part\_measured

The `part_measured` specifies the `Sampled_set` (see 4.2.275) that measures the properties of the parts produced by the `Activity_execution_result`. The `part_measured` need not be specified for a particular `Activity_execution_result`. See 4.3.13 for the application assertion.

## 4.2.3 Activity\_result\_record

An `Activity_result_record` is a log of how a manufacturing operation was carried out. Each `Activity_result_record` is either a `Equipment_setting_record` (see 4.2.112), a `Process_parameter_record` (see 4.2.235), a `Material_usage_record` (see 4.2.191), or a `Composition_element_record` (see 4.2.38).

NOTE `Activity_result_record` objects, in combination with inspection results, can be used by techniques such as statistical process control, to determine the relationship between quality factors and process variables, thus enabling continuous improvement of product quality.

The data associated with a `Activity_result_record` are the following:

— `date_and_time`.

### 4.2.3.1 date\_and\_time

The `date_and_time` specifies the day and time at which the `Activity_result_record` was made. The `date_and_time` need not be specified for a particular `Activity_result_record`.

NOTE This would normally be when the actual process parameters are measured and recorded during production.

## 4.2.4 Address

An `Address` is a person or organization defined address for mail delivery. The data associated with an `Address` are the following:

— `city`;

— `e_mail`;

— `fax`;

— `state`;

ISO 10303-223:2008 (E)

- street\_address;
- web\_url;
- zip\_code.

#### **4.2.4.1 city**

The city specifies the name of a town.

#### **4.2.4.2 e\_mail**

The e\_mail specifies the electronic mail address at which electronic mail may be received.

#### **4.2.4.3 fax**

The fax specifies the number at which facsimiles may be received.

#### **4.2.4.4 state**

The state specifies the name of a region in a country.

#### **4.2.4.5 street\_address**

The street\_number specifies the number of a location on a street.

#### **4.2.4.6 web\_url**

The web\_url specifies the World Wide Web website Uniform Resource Locator for a person or organization.

#### **4.2.4.7 zip\_code**

The zip\_code specifies the code that is used by the country's postal service.

### **4.2.5 Allowed\_time**

An Allowed\_time is a type of Performance\_rate (see 4.2.225) that is the total time allocated for the completion of an activity.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.2.

EXAMPLE The Allowed\_time equals the allowance\_factor multiplied by the standard\_time. If the standard\_time takes 3 minutes to drill a hole, and the allowance\_factor for the job is 1.2, the Allowed\_time for the job would be 3.6 minutes.

The data associated with an `Allowed_time` are the following:

- `allowance_factor`;
- `allowed_type`;
- `standard_time`.

#### **4.2.5.1 allowance\_factor**

The `allowance_factor` specifies the percentage of the standard time which compensates for unavoidable delays.

EXAMPLE Unavoidable delays include such thing as: making minor tool repairs, equipment repairs, operator fatigue, and human factors.

#### **4.2.5.2 allowed\_type**

The `allowed_type` specifies the word or group of words that identify the item within the category of allowed times. An `Allowed_time` need not be specified for a particular `allowed_type`.

EXAMPLE The type includes such words as `batch`, `lot`, `order`, `piece`, `setup`, `teardown`, and `run time`.

#### **4.2.5.3 standard\_time**

The `standard_time` specifies a time value for the accomplishment of a work task as determined by the proper application of appropriate work measurement techniques.

NOTE The `standard_time` is not influenced by operator fatigue, weather conditions, or other variables.

### **4.2.6 Alternate\_activity**

An `Alternate_activity` is a type of `Activity` (see 4.2.1) that is the identification of a separate activity that can be used to replace the current activity within the same process plan.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.3.

The data associated with an `Alternate_activity` is the following:

- `primary_activity`.

#### **4.2.6.1 primary\_activity**

The `primary_activity` specifies the `Activity` (see 4.2.1) of the primary activity for which this activity is the `Alternate_activity`. See 4.3.14 for the application assertion.

## **4.2.7 Alternate\_process\_plan**

An `Alternate_process_plan` is the identification of a secondary process plan that may be used when the primary process plan choice is not available. The data associated with an `Alternate_process_plan` is the following:

- `alternate_ranking`;
- `plan_substitute`.

### **4.2.7.1 alternate\_ranking**

The `alternate_ranking` specifies the order for selecting `Alternate_process_plan` objects in the event the primary `Process_plan_version` (see 4.2.238) is not available.

### **4.2.7.2 plan\_substitute**

The `plan_substitute` specifies `Process_plan_version` (see 4.2.238) that is to be used as a substitute process plan. See 4.3.15 for the application assertion.

## **4.2.8 Ancillary\_activity**

An `Ancillary_activity` is a type of `Casting_activity` (see 4.2.21) that defines an ancillary action.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.6.

The data associated with an `Ancillary_activity` are the following:

- `method`.

### **4.2.8.1 method**

The `method` specifies the descriptive ancillary action being performed. The `method` need not be specified for a particular `Ancillary_setup`.

## **4.2.9 Ancillary\_setup**

An `Ancillary_setup` is a type of `Activity_setup` (see 4.2.280) that defines the setup for a an ancillary action `Activity` (see 4.2.1).

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.7.

The data associated with an `Ancillary_setup` are the following:

- `setup_type`.

### 4.2.9.1 setup\_type

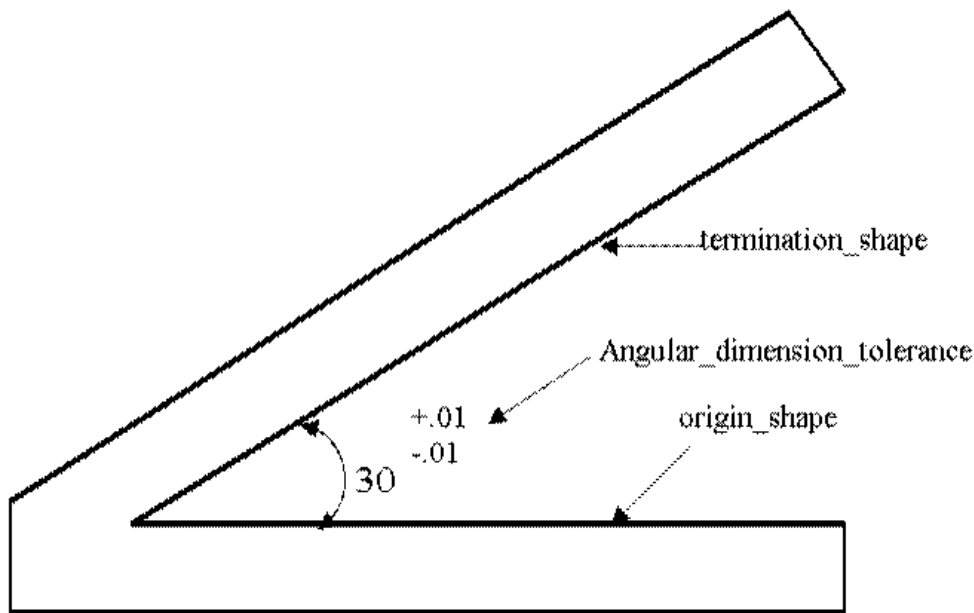
The setup\_type specifies the descriptive placement and location information for an ancillary activity.

### 4.2.10 Angular\_dimension\_tolerance

An Angular\_dimension\_tolerance is a type of Location\_tolerance (see 4.2.169) that defines the allowable variation in the angle between two elements of the shape of a part. Each Angular\_dimension\_tolerance shall have an origin\_shape and a termination\_shape.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.3.

NOTE 2 Figure 3 illustrates the Angular\_dimension\_tolerance.



**Figure 3 — Angular\_dimension\_tolerance**

The data associated with an Angular\_dimension\_tolerance are the following:

— plane\_and\_direction.

### 4.2.10.1 plane\_and\_direction

The `plane_and_direction` specifies a plane that contains the geometry for the `Angular_dimension_tolerance` and a direction that is the location of the plane that contains the `Angular_dimension_tolerance`. The `plane_and_direction` need not be specified for a particular `Angular_dimension_tolerance`. See 4.3.16 for the application assertion.

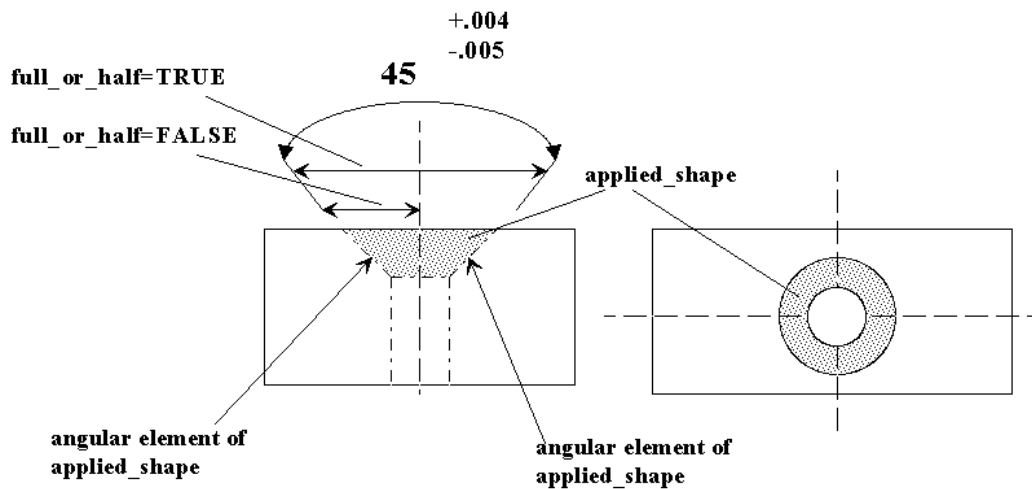
EXAMPLE A part might be viewed in a front view for defining a `location_dimension_tolerance`.

### 4.2.11 Angular\_size\_dimension\_tolerance

An `Angular_size_dimension_tolerance` is a type of `Size_tolerance` (see 4.2.304) that specifies the allowable variation on the size or gap formed by two angular elements of the shape of a part.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.4.

NOTE 2 Figure 4 illustrates the `Angular_size_dimension_tolerances`.



**Figure 4 — Angular\_size\_dimension\_tolerance**

The data associated with an `Angular_size_dimension_tolerance` are the following:

- `full_or_half`;
- `major_angle`.

#### 4.2.11.1 full\_or\_half

The `full_or_half` specifies a Boolean value that indicates the method used to establish the `Angular_size_dimension_tolerance` angle. A value of true specifies the angle is established between the two sides of an angular element. A value of false specifies the angle is established between a center line datum and an angular element.

#### 4.2.11.2 major\_angle

The `major_angle` specifies a Boolean value that indicates the size of the angle for defining the variation. A value of TRUE specifies the angle to be the larger of the two angles formed by the two elements of the part's shape that are related to the `Angular_size_dimension_tolerance`.

### 4.2.12 Approval

An Approval is the indication within an organization of concurrence or nonconcurrence with a piece of product data.

NOTE This application object is harmonized with the same object from 10303-224:2006, Clause 4.2.6.

The data associated with an Approval are the following:

- `approval_authority`;
- `approval_date`;
- `status`.

#### 4.2.12.1 approval\_authority

The `approval_authority` specifies the responsible person and company for making the approval. There may be more than one `approval_authority` for an Approval. See 4.3.18 for the application assertion.

#### 4.2.12.2 approval\_date

The `approval_date` specifies a calendar date on which the approval status was set. See 4.3.17 for the application assertion.

#### 4.2.12.3 status

The `status` specifies the state of consent applied to the approval to manufacture a part.

The values of the status may be one of the following:

- approved;
- disapproved;
- not\_yet\_approved;
- withdrawn.

NOTE See 4.2.12.3.1 to 4.2.12.3.4 for the definition of each allowable value for status type.

**4.2.12.3.1 approved:** a concurrence has been given on a specific date.

**4.2.12.3.2 disapproved:** a nonconcurrence has been given on a specific date.

**4.2.12.3.3 not\_yet\_approved:** a concurrence has not yet been given.

**4.2.12.3.4 withdrawn:** a concurrence has been removed on a specific date.

## 4.2.13 Assembly

An Assembly is a type of Non\_casting\_operations (see 4.2.202) that identifies the components that are put together to form a desired assembly. The data associated with an Assembly are the following:

- foam\_pattern\_coating\_density;
- pattern\_section\_assembly\_sequence.

### 4.2.13.1 foam\_pattern\_coating\_density

The foam\_pattern\_coating\_density specifies the weight per unit volume of refractory material used to coat the foam pattern.

### 4.2.13.2 pattern\_section\_assembly\_sequence

The pattern\_section\_assembly specifies the sequence of foam sections to be assembled and glued for a specified pattern shape to produce the cast part.

## 4.2.14 Bankout\_frame

A Bankout\_frame is a type of Core\_equipment (see 4.2.54) that is used to retain dry sand in which a fragile core is embedded for support. The data associated with an Bankout\_frame are the following:

- material\_type.



#### 4.2.14.1 material\_type

The **material\_type** specifies the word or group of words that describes the substance from which the **Bankout\_frame** is made.

#### 4.2.15 Base\_shape

A **Base\_shape** is the bounded geometry for the part before machining features are applied. The **Base\_shape** defines the form and minimum dimensions required to determine the raw stock (material) required to be issued for the manufacture of a part. Each **Base\_shape** may be a **Casting\_envelope\_**-requirement (see 4.2.23)

#### 4.2.16 Bench\_inspection

**Bench\_inspection** is a type of **Inspection\_activity** (see 4.2.153) that is a manual procedure for evaluating the dimensional accuracy of a cast part against the design specification for every dimension with is associated tolerance for the part.

#### 4.2.17 Boundary\_condition

A **Boundary condition** is the state of physical phenomena at the surface of a region.

NOTE **Boundary conditions** include heat transfer coefficients and pressure on mould and vents.

The data associated with a **Boundary\_condition** are the following:

- **applies\_to**;
- **condition\_value**.

##### 4.2.17.1 applies\_to

The **applies\_to** specifies the surface to which the **Boundary condition** pertains. See 4.3.21 for the application assertion.

##### 4.2.17.2 condition\_value

The **condition\_value** specifies the property value or relationship between property values that characterizes the **Boundary\_condition**. See 4.3.20 for the application assertion.

#### 4.2.18 BSU

A **BSU** identifies a piece of information, that can be a supplier, a class or a property, by specifying a code and a version. Each **BSU** is either a **Supplier\_BSU** (see 4.2.315), a **Class\_BSU** (see 4.2.35), or a **Property\_BSU** (see 4.2.254).

ISO 10303-223:2008 (E)

The data associated with a BSU are the following:

— code.

#### **4.2.18.1 code**

The code specifies the designation of the identification of the information piece.

#### **4.2.19 Cast\_part**

A Cast\_part is a type of Part\_version (see 4.2.217) that is a metal object formed by a casting process to the required shape as distinct from one shaped by a mechanical process.

The data associated with a Cast\_part are the following:

— cast\_feature.

##### **4.2.19.1 cast\_feature**

The cast\_feature specifies the types of features necessary to manufacture a cast part. See 4.3.22 for the application assertion.

#### **4.2.20 Cast\_part\_with\_rigging**

A Cast\_part\_with\_rigging is a type of Part\_version (see 4.2.217) that is a casting just after it has been removed from the mould and before there has been an attempt to remove gating, sprues, and risers.

The data associated with a Cast\_part\_with\_rigging are the following:

— cast\_feature;

— rigging

##### **4.2.20.1 cast\_feature**

The cast\_feature specifies the types of features necessary to manufacture a cast part. See 4.3.23 for the application assertion.

##### **4.2.20.2 rigging**

The rigging specifies the necessary components of a Gating\_system (see 4.2.134) to produce a cast part. See 4.3.24 for the application assertion.

## 4.2.21 Casting\_activity

A Casting\_activity is a type of Activity (see 4.2.1) that define any activity that interacts with the part being cast. Each Casting\_activity is either a Ancillary\_activity (see 4.2.8), General\_foundry\_process (see 4.2.26), Non\_casting\_operations (see 4.2.202), Non\_permanent\_moulding\_process (see 4.2.204), Inspection\_activity (see 4.2.153), Permanent\_moulding\_process (see 4.2.226), Simulation\_process (see 4.2.295) or Tooling\_process (see 4.2.334).

NOTE This application object is harmonized with the manufacturing\_activity object from ISO 10303-240:2005, Clause 4.2.76.

The data associated with a Manufacturing\_activity are the following:

— micro\_plan\_reference.

### 4.2.21.1 micro\_plan\_reference

The micro\_plan\_reference specifies a reference to an external ISO standard that contains information defining a micro process plan that supports this manufacturing activity. This micro process plan shall be an ISO 10303 part. The micro\_plan\_reference need not be specified for a particular Casting\_activity. See 4.3.25 for the application assertion.

NOTE A valid micro process plan may be represented as ISO 10303-238

## 4.2.22 Casting\_design\_feature

A Casting\_design\_feature is a type of Feature (see 4.2.124) that identifies the types of features necessary to manufacture a cast part. Each Casting\_design\_feature is either a Manufacturing\_design\_feature (see 4.2.176) Fixture\_tab (see 4.2.130) or a Rib (see 4.2.271). The data associated with a Casting\_design\_feature are the following:

- draft;
- machine\_stock;
- placement.

### 4.2.22.1 draft

The draft specifies the angle that defines the taper on vertical elements in a pattern which allows easy withdrawal of the pattern from the mould. See 4.3.27 for the application assertion.

NOTE The amount of draft required will vary with the depth of the pattern.

#### **4.2.22.2 machine\_stock**

The `machine_stock` specifies the additional material added to the cast part to allow for machining. See 4.3.26 for the application assertion.

#### **4.2.22.3 placement**

The `placement` specifies the position and orientation of a `Casting_design_feature` relative to the base shape for a part. See 4.3.28 for the application assertion.

### **4.2.23 Casting\_envelope\_requirement**

A `Casting_envelope_requirement` is a type of `Base_shape` (see 4.2.15) that is the minimum dimensional space required for the cast part to occupy. The data associated with a `Casting_envelope_requirement` are the following:

- `part_length`;
- `part_height`;
- `part_width`.

#### **4.2.23.1 part\_length**

The `part_length` specifies the longest extent of the cast part as a dimensional numeric value with units. See 4.3.29 for the application assertion.

#### **4.2.23.2 part\_height**

The `part_height` specifies the dimensional height or depth as a numeric value with units of the cast part. See 4.3.29 for the application assertion.

#### **4.2.23.3 part\_width**

The `part_width` specifies the dimensional breadth or width as a numeric value with units of the cast part. See 4.3.29 for the application assertion.

### **4.2.24 Casting\_insert**

A `Casting_insert` is a type of `Sand_casting_design_feature` (see 4.2.276) that is a secondary metallic material that is placed in the mould and is cast into the body of the casting which then becomes an integral part of the final casting.

NOTE This application object is harmonized with the part insert object from ISO 8062-1:2007, Clause 2.9.

The data associated with a Casting\_insert are the following:

- insert\_name;
- material\_type;
- positioning\_method.

#### **4.2.24.1 insert\_name**

The insert\_name specifies a label or description that is associated with the function of the insert in the casting. The insert\_name need not be specified for a particular Casting\_insert.

#### **4.2.24.2 material\_type**

The material\_type specifies the description of the material used for the insert.

#### **4.2.24.3 positioning\_method**

The positioning\_method specifies the way or style of locating the insert in the mould before the mould is closed.

### **4.2.25 Casting\_per\_order**

A Casting\_per\_order is a type of Casting\_verification (see 4.2.28) that is the required number of production castings per order to be evaluated for compliance to the design record and required specifications. The data associated with a Casting\_per\_order are the following:

- sampling\_of\_parts.

#### **4.2.25.1 sampling\_of\_parts**

The sampling\_of\_parts specifies the requirements for a specified number of production run cast parts to be tested and inspected. See 4.3.30 for the application assertion.

EXAMPLE The sample could be one cast part from every 10 production castings made or a sample could be a specified quantity from each shift or from each furnace melt or from each inoculation.

### **4.2.26 General\_foundry\_process**

A General\_foundry\_process is a type of Casting\_activity (see 4.2.21) that is that define the processing activity to interact with the material part being cast.

ISO 10303-223:2008 (E)

The data associated with a `General_foundry_process` are the following:

- `core_making_process`;
- `core_to_be_made`;
- `described_by`;
- `identifies`;
- `mismatch_allowances`;
- `process_parameter`;
- `type_of_operation`.

#### **4.2.26.1 core\_making\_process**

The `core_making_process` specifies the method used to produce the core. A `core_making_process` need not be specified for a particular `General_foundry_process`.

NOTE See 4.2.26.1.1 to 4.2.26.1.5 for the definition of each allowable value for `core_process_type`.

**4.2.26.1.1 shell:** A process for forming a mould from resin-bonded sand mixtures brought in contact with preheated metal patterns, resulting in a firm shell with a cavity corresponding to the outline of the pattern.

**4.2.26.1.2 green\_sand:** A process to produce a core in the green sand state or a body of sand formed directly from a pattern in making the mould. A core that is not baked.

**4.2.26.1.3 no\_bake:** Moulds and cores produced with a resin-bonded air-setting sand. Also known as the air set process because moulds are left to harden under atmospheric conditions.

**4.2.26.1.4 lost\_foam:** A casting process in which a foam pattern is placed in a flask and filled with loose sand to form a cast part.

**4.2.26.1.5 gas\_cured:** a core produced with a cold box system with a SO<sub>2</sub> or CO<sub>2</sub> process with resin bonded sand that is cured with a vaporized gas catalyst.

#### **4.2.26.2 core\_to\_be\_made**

The `core_to_be_made` specifies the `Production_core_box` requirements to produce the required core for the cast part. A `core_to_be_made` need not be specified for a particular `General_foundry_process`. See 4.3.35 for the application assertion.

### 4.2.26.3 described\_by

The `described_by` specifies the `Shape_aspect` (see 4.2.282) that defines a portion of the part shape being affected by the `Casting_activity`. A `described_by` need not be specified for a particular `General_foundry_process`. A `described_by` need not be specified for a particular `Shape_aspect`. See 4.3.36 for the application assertion.

EXAMPLE An example is the geometric shape of a pocket that will be the result of a process being performed.

### 4.2.26.4 identifies

The `identifies` specifies the `Equipment_setup` (see 4.2.113) that defines the positioning of the tooling for the part as located on the equipment for the process being affected by the `Casting_activity`. An `identifies` need not be specified for a particular `General_foundry_process`. See 4.3.31 for the application assertion.

EXAMPLE An example is the geometric shape of a pocket that will be the result of a process being performed.

### 4.2.26.5 process\_parameter

The `process_parameter` specifies the `Property_parameter` (see 4.2.256) that define additional process information. See 4.3.33 and 4.3.34 for the application assertion.

### 4.2.26.6 mismatch\_tolerances

The `mismatch_allowance` specifies the maximum mismatch value permitted for a cast part during any required casting process. A `mismatch_tolerances` need not be specified for a particular `General_foundry_process`. See 4.3.32 for the application assertion.

### 4.2.26.7 type\_of\_operation

The `type_of_operation` specifies the type of process being performed.

NOTE See 4.2.26.7.1 to 4.2.26.7.24 for the definition of each allowable value for type of operation.

**4.2.26.7.1 core\_presentation:** the process of a core being located in a die mould.

**4.2.26.7.2 core\_setting:** the process of a core being located in a die mould.

**4.2.26.7.3 die\_assembly:** the process of assembling component parts of a die to form the die mould.

**4.2.26.7.4 die\_handling:** the process of moving the die components for assembly or the moving of a die mould from one location to another.

**4.2.26.7.5 die\_temperature\_control:** the process of controlling the distribution and variation of temperature in or on the die mould.

**4.2.26.7.6 furnace\_loading:** the process of placing the required solid metals and other materials into the melting furnace sometimes called charging.

**4.2.26.7.7 gate\_system\_removal:** the process of removing the ingates and risers from the cast part.

**4.2.26.7.8 grinding:** the process of removing the flash, ingate contacts, riser contacts, and dressing the partings of a cast part by use of a grinder.

**4.2.26.7.9 inspection:** the process of inspecting the cast part for flaws, design specification verification, and tooling accuracy.

**4.2.26.7.10 ladle\_loading:** the process of filling the pouring or holding ladle with molten metal.

**4.2.26.7.11 loading\_injection\_system:** the process of filling the sprue or shot sleeve with molten metal.

**4.2.26.7.12 melt\_raw\_material:** the process of bringing solid raw materials to a molten or fluid state.

**4.2.26.7.13 mould\_closing:** the process of assembling and placing all required item in a mould, bringing mating components of the mould together, and locking the mould for filling.

**4.2.26.7.14 mould\_making\_process:** the process of making a permanent or non-permanent mould.

**4.2.26.7.15 package\_and\_shipping:** the process of preparing the cast part to ship and delivery.

**4.2.26.7.16 part\_handling:** the action or process of handling and moving the cast part from one operation or location to another.

**4.2.26.7.17 part\_shakeout:** the operation of separating the casting from the mould material.

NOTE The mould material is usually from a sand mould.

**4.2.26.7.18 tool\_handling:** the action or process of handling and moving the pattern and core box tooling from one operation or location to another.

**4.2.26.7.19 pouring\_process:** the process of moving the filled ladle to each mould and filling each mould.

**4.2.26.7.20 refractor\_removal:** the process of the removing the refractory substance from a cast part.

**4.2.26.7.21 removal\_from\_die:** the process of removing a pattern, core, or casting from a metal die.

**4.2.26.7.22 sand\_removal:** the process of removing attached sand from the casting after shakeout.

**4.2.26.7.23 slurry\_process:** the process of adding sufficient refractory material to a wax or foam pattern structure for receiving molten metal.

**4.2.26.7.24 wax\_removal:** the process of removing wax pattern materials from an investment tree.



## 4.2.27 Casting\_service\_data

A Casting\_per\_order is a type of Customer\_casting\_requirement (see 4.2.62) that is the information that describes the requirements and types of usage application of the cast part in service. The Casting\_service\_data also describes the additional operations that are required for the cast part.

The data associated with a Casting\_service\_data are the following:

- comments;
- corrosion\_service\_requirement;
- elevated\_temperature\_service;
- machining\_and\_finishing\_requirements.;
- mechanical\_stress\_requirements;
- subject\_to\_abrasive\_wear\_by;
- subject\_to\_wear\_against;
- wear\_or\_abrasion\_lubrication.

### 4.2.27.1 comments

The comments specifies specific parameter data for the required service the cast part is designed to withstand. The comments need not be specified for a particular Casting\_service\_data for the cast part. See 4.3.38 for the application assertion.

### 4.2.27.2 corrosion\_service\_requirement

The corrosion\_service\_requirement specifies the type of corrosive material and applications with numerical data for which the cast part is designed. The corrosion\_service\_requirement need not be specified for a particular Casting\_service\_data for the cast part. See 4.3.37 for the application assertion.

### 4.2.27.3 elevated\_temperature\_service

The elevated\_temperure\_service specifies the type of increased temperatures with numerical data for which the cast part is designed. The elevated\_temperature\_service need not be specified for a particular Casting\_service\_data for the cast part. See 4.3.39 for the application assertion.

#### **4.2.27.4 machining\_and\_finishing\_requirements**

The `mechanical_stress_requirement` specifies the type of stress with its numerical data for which the cast part is designed. The `mechanical_stress_requirement` need not be specified for a particular `Casting_service_data` for the cast part. See 4.3.40 for the application assertion.

#### **4.2.27.5 mechanical\_stress\_requirements**

The `mechanical_stress_requirement` specifies the type of stress with its numerical data for which the cast part is designed. The `mechanical_stress_requirement` need not be specified for a particular `Casting_service_data` for the cast part. See 4.3.42 for the application assertion.

#### **4.2.27.6 subject\_to\_abrasive\_wear\_by**

The `subject_to_abrasive_wear_by` specifies the Material that will cause deterioration or erosion of the cast part. The `subject_to_abrasive_wear_against` need not be specified for a particular `Casting_service_data` for the cast part. See 4.3.41 for the application assertion.

#### **4.2.27.7 subject\_to\_wear\_against**

The `subject_to_wear_against` specifies the Material that may cause wear or fatigue conditions on the cast part. The `subject_to_wear_against` need not be specified for a particular `Casting_service_data` for the cast part. See 4.3.41 for the application assertion.

#### **4.2.27.8 wear\_or\_abrasion\_lubrication**

The `wear_with_lubrication` specifies the degree of wear when lubrication is used with the material identified for `subject_to_wear_against`. The `wear_with_lubrication` need not be specified for a particular `Casting_service_data` for the cast part. See 4.3.38 for the application assertion.

Note this attribute is only valid when the `subject_to_wear_against` specifies a Material.

The values of the `wear_with_lubrication` may be one of the following:

- good;
- intermittent;
- none.

NOTE See 4.2.27.8.1 to 4.2.27.8.3 for the definition of each allowable value for wear type.

**4.2.27.8.1 good:** the wear on the cast part is good when lubrication is used.

**4.2.27.8.2 intermittent:** there is some wear on the cast part when lubrication is used.

**4.2.27.8.3 none:** there is no wear on the cast part when lubrication is used.

## 4.2.28 Casting\_verification

A Casting\_verification is the documented proof that a cast part has satisfied or failed to satisfy all design record requirements and specifications for the part as designed. Each Casting\_verification is either a first\_article (see 4.2.129) or a casting\_per\_order (see 4.2.25).

The data associated with a Casting\_verification are the following:

- cavity\_impression\_id;
- destruction;
- nde;
- verify\_with.

### 4.2.28.1 cavity\_impression\_id

The cavity\_impression\_id identifies which cavity or impression produced the moulded for the cast part.

### 4.2.28.2 destruction

The destruction specifies a destructive test that is applied to the cast part to certify the part has met all quality requirements. A destruction need not be specified for a particular Casting\_verification for the cast part. See 4.3.43 for the application assertion.

EXAMPLE visual, dimensional, x-ray, pressure, ultrasonic, etc. are inspection and test procedures that are non-destructive testing often known as NDT procedures.

### 4.2.28.3 nde

The nde specifies one or more non-destructive evaluations that are applied to the cast part to certify the part has met all quality requirements. See 4.3.43 for the application assertion.

EXAMPLE visual, dimensional, x-ray, pressure, ultrasonic, etc. are inspection and test procedures that are non-destructive testing often known as NDT procedures.

### 4.2.28.4 verify\_with

The verify\_with specifies one or more tolerance requirements to be used for casting verification. See 4.3.43 for the application assertion.

## 4.2.29 Casting\_weight\_requirement

A Casting\_weight\_requirement is a type of Base\_shape (see 4.2.15) that defines the weight value with appropriate units for the cast part. The data associated with a Casting\_verification are the following:

- property\_characteristic;
- weight\_declared.

### 4.2.29.1 property\_characteristic

The property\_characteristic specifies the type of weight declared for the cast part. A property\_characteristic need not be specified for a particular Casting\_weight\_requirement for the cast part. See 4.3.45 for the application assertion.

### 4.2.29.2 weight\_declared

The weight\_declared specifies the weight as actual or estimated.

The values of the weight\_declared may be one of the following:

- actual;
- estimated.

NOTE See 4.2.29.2.1 to 4.2.29.2.2 for the definition of each allowable value for weight\_declared type.

**4.2.29.2.1 actual:** is the actual measured weight of the cast part for shipment.

**4.2.29.2.2 estimated:** is the approximate weight of the cast part for shipment. The estimated weight is a calculated value based on metal volume.

## 4.2.30 Chaplet

A Chaplet is a type of Sand\_casting\_design\_feature (see 4.2.276) that is a metal support that stabilizes the core in the mould when the core seat or the core strength is inadequate to support the core.

NOTE 1 Figure 5 illustrates the Chaplet.

NOTE 2 This application object is harmonized with the chaplet definition from glossary of metalcasting terms from AFS [2].

The data associated with a Chaplet are the following:

- chaplet\_size;
- chaplet\_type;
- pad.

#### 4.2.30.1 chaplet\_size

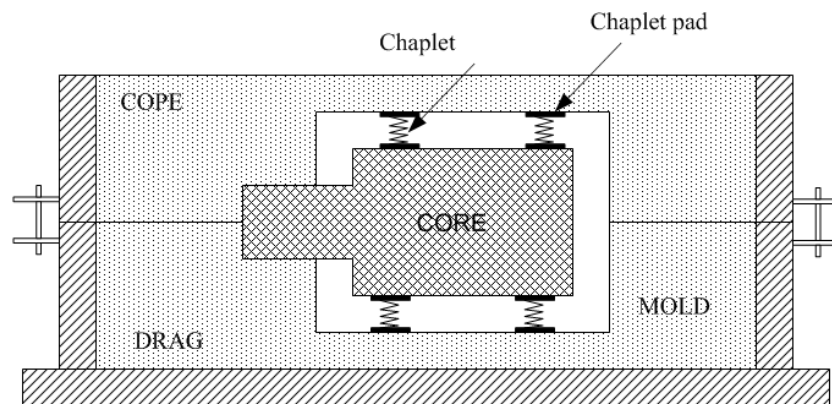
The chaplet\_size specifies the dimensional size of the chaplet to be used in the mould. See 4.3.47 for the application assertion.

#### 4.2.30.2 chaplet\_type

The chaplet\_type specifies the kind of chaplet to be used in the mould to support the core.

#### 4.2.30.3 pad

The pad specifies the size and location of each Chaplet\_pad attached to the Production\_pattern or Core\_box. A pad need not be specified for a particular Chaplet. See 4.3.46 for the application assertion.



**Figure 5 — Chaplet and Chaplet\_pad**

### 4.2.31 Chaplet\_pad

A Chaplet\_pad is a type of Sand\_casting\_design\_feature (see 4.2.276) that is a shallow projection attached to the cope and drag surfaces of a pattern and core box that are opposite of the parting surfaces.

NOTE 1 Each chaplet pad will make an impression in the mould sand or core sand to provide a seat to set a chaplet.

NOTE 2 Figure 5 illustrates the Chaplet with Chaplet\_pad.

The data associated with a Chaplet\_pad are the following:

— pad\_size.

#### 4.2.31.1 pad\_size

The pad\_size specifies the dimensional size values for each chaplet\_pad attached to the pattern or core box. See 4.3.48 for the application assertion.

### 4.2.32 Checking\_aid\_tool

A Checking\_aid\_tool is a device or an implement used during the inspection process that provides assistance in the checking procedures. The data associated with a Checking\_aid\_tool are the following:

— aid\_id;

— aid\_name;

— aid\_tool;

— revision\_date;

— revision\_id.

#### 4.2.32.1 aid\_id

An aid\_id specifies the alphanumeric label that identifies the aid.

#### 4.2.32.2 aid\_name

An aid\_name specifies the alphanumeric title or term used to describe the aid.

### 4.2.32.3 aid\_tool

The aid\_tool specifies the device or implement that was used to assist in the inspection. The values of the aid\_tool may be one of the following:

- paper\_document;
- gauge;
- fixture.

NOTE See 4.2.32.3.1 to 4.2.32.3.4 for the definition of each allowable value for aid\_tool.

**4.2.32.3.1 fixture:** a device or implement used to locate, position, or hold securely an item to be inspected or tested.

**4.2.32.3.2 gauge:** a device or implement used to measure, locate, or position an item to be inspected or tested.

**4.2.32.3.3 paper\_document:** a certificate, reference, or form used to validate an item during the inspection or testing process.

**4.2.32.3.4 template:** a device or implement used to measure or locate an item to be inspected or tested.

### 4.2.32.4 revision\_date

The revision\_date specifies the month, day, and year for the revision of the aid used during the inspection procedure. See 4.3.49 for the application assertion.

### 4.2.32.5 revision\_id

An revision\_id specifies the alphanumeric label that identifies the revision level of the aid.

## 4.2.33 Chill

A Chill is a type of Sand\_casting\_design\_feature (see 4.2.276) that is is a specific area of a casting that requires a harder surface or solidification control.

NOTE 1 A metallic chill insert is placed in the mould to promote rapid cooling at a desired point to make the casting harder at this point or to promote directional solidification and prevent shrinkage at a hot spot which cannot be fed from a riser. add the rest of the highlighted text for the note, but delete what is circled in blue.

NOTE 2 This application object is harmonized with the chill object from ISO 8062-1:2007, Clause 3.7.

ISO 10303-223:2008 (E)

The data associated with a Chill are the following:

- chill\_area\_shape;
- internal\_or\_external\_chill;
- material\_type;

#### **4.2.33.1 chill\_area\_shape**

The chill\_area\_shape specifies the form and configuration of the external chill shape. See 4.3.50 for the application assertion.

NOTE There are two types of chills:

- internal chills are chills that when used become part of the finished casting.
- external chills are chills that are imbedded in the mould and are reusable.

#### **4.2.33.1.1 internal\_or\_external\_chill**

An internal\_or\_external\_chill specifies the placement of each chill. Each chill will be positioned internally to be apart of the finished casting or be placed externally as part of the mould or core and to be reusable.

#### **4.2.33.2 material\_type**

The material\_type specifies the type of material used for the chill.

NOTE Material for the internal chill is to be of the basic composition as the metal required for the cast part as the chill will be an integral component of the finished casting. The material for an external chill is to be of a material with a higher melting point than the material used for the cast part so as to prevent fusion to the cast part.

### **4.2.34 Choke**

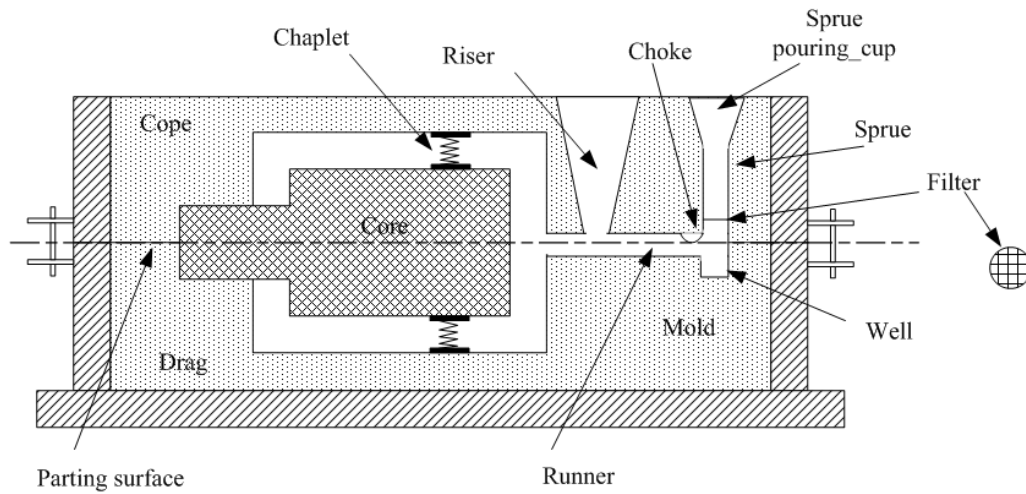
The Choke is a type of Gating\_design\_feature (see 4.2.133) that is the part of the gating system that most restricts or regulates the flow of metal into the mould cavity.

NOTE Figure 6 illustrates a Gating\_system with a Choke.

The data associated with a Choke are the following:

- choke\_location;
- choke\_size.





**Figure 6 — Choke connection in a Gating\_system**

#### 4.2.34.1 choke\_location

The `choke_location` specifies the placement of a Choke in the Runner. See 4.3.52 for the application assertion.

#### 4.2.34.2 choke\_size

The `choke_size` specifies the dimensional size values of the Choke. See 4.3.51 for the application assertion.

#### 4.2.35 Class\_BSU

A `Class_BSU` is a type of `BSU` (see 4.2.18) that identifies a class in a parts library.

NOTE 1 `BSU` is an acronym for “basic semantical unit”.

NOTE 2 The combination of supplier identification, code, and version of a class is to be unique.

The data associated with a `Class_BSU` are the following:

- `defined_by`;
- `version`.

ISO 10303-223:2008 (E)

#### **4.2.35.1 defined\_by**

The `defined_by` specifies the library supplier who defines the library class. See 4.3.53 for the application assertion.

#### **4.2.35.2 version**

The `version` specifies the designation of the version of the information piece.

#### **4.2.36 Common\_datum**

A `Common_datum` is a type of `Datum` (see 4.2.69) that is a set of two or more `Datum_feature` objects which are for establishing a single datum plane or axis.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.38.

The data associated with a `Common_datum` are the following:

— `element`.

#### **4.2.36.1 element**

The `element` specifies the list of `Datum_feature` objects for defining the `Common_datum`. See 4.3.54 for the application assertion.

#### **4.2.37 Composition\_element**

A `Composition_element` is the amount of an element in a `Material` (see 4.2.183). The data associated with a `Composition_element` are the following:

— `amount`

— `chemical_element`.

#### **4.2.37.1 amount**

The `amount` specifies the quantitative measure of a chemical element in a `Material`.

#### **4.2.37.2 chemical\_element**

The `chemical_element` specifies the chemical element that is contained in a `Material` in the amount specified by the `amount` attribute.

### 4.2.38 Composition\_element\_record

A `Composition_element_record` is a type of `Activity_result_record` (see 4.2.3) a record of the quantitative measure of a chemical element in a `Material` (see 4.2.183) during a particular execution of a `Activity` (see 4.2.1).

NOTE A `Composition_element_record` record may be used for later analysis to determine the effect of chemical composition on the quality of the part.

The data associated with a `Composition_element_record` record are the following:

- `actual_composition_amount`;
- `recorded_usage`.

#### 4.2.38.1 actual\_composition\_amount

The `actual_substance_composition_amount` specifies the measured amount of a chemical element in a part during a particular execution of an activity. See 4.3.56 for the application assertion.

#### 4.2.38.2 recorded\_usage

The `recorded_usage` specifies the `Composition_element` (see 4.2.37) within a process whose actual value is recorded by the `Composition_element_record`. See 4.3.55 for the application assertion.

### 4.2.39 Composition\_inspection

A `Composition_inspection` is a type of `Inspection_or_test_result` (see 4.2.155) that is the outcome of an examination or measurement of a material. The data associated with a `Composition_inspection` are the following:

- `composition_result`;
- `requirement`

#### 4.2.39.1 composition\_result

The `composition_result` specifies the outcome of the `Composition_inspection`. See 4.3.57 for the application assertion.

#### 4.2.39.2 requirement

The `requirement` specifies the `Material` (see 4.2.183) that was evaluated to produce the `Composition_inspection`. See 4.3.58 for the application assertion.

## 4.2.40 Compound\_feature

A Compound\_feature is a type of Machining\_feature (see 4.2.177) that is a union of one or more Machining\_feature objects to create a more complex feature definition. The placement of a Compound\_feature is relative to either the part, another Compound\_feature, or a Replicate\_feature which uses a Compound\_feature as the base feature. Features which are elements of the Compound\_feature have placement defined relative to the Compound\_feature placement.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.239.

NOTE 2 Figure 7 illustrates an example of a Compound\_feature with one counterbore\_hole feature and one countersunk\_hole feature.

The data associated with a Compound\_feature are the following:

- element;
- feature\_description;
- feature\_name.

### 4.2.40.1 element

The element specifies the base features that when combined defines a Compound\_feature. There shall be more than one element for a Compound\_feature. See 4.3.59 for the application assertion.

### 4.2.40.2 feature\_description

The feature\_description specifies a user defined explanation of the Compound\_feature. A Compound\_feature may but need not require a feature\_description.

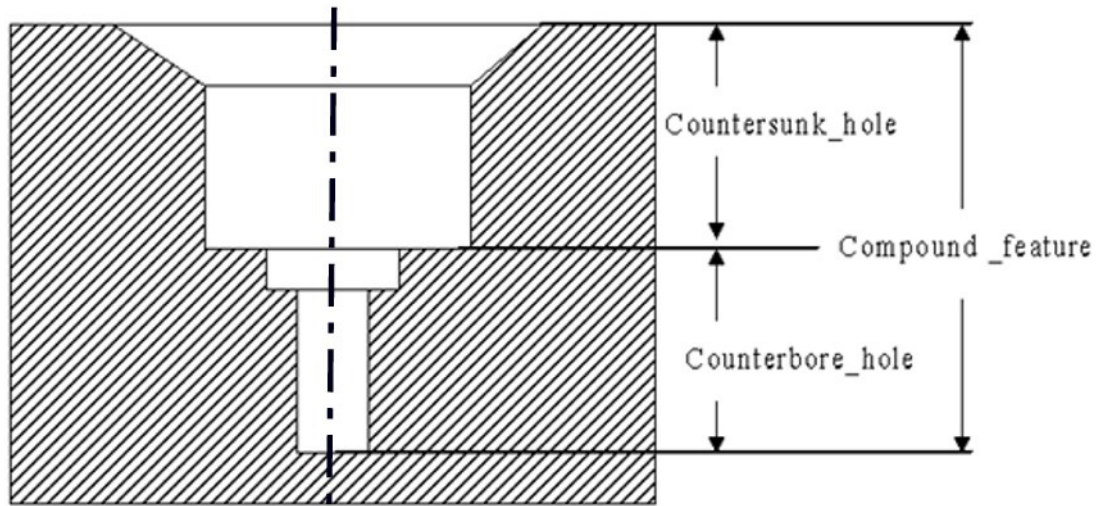
### 4.2.40.3 feature\_name

The feature\_name specifies a user defined name for the Compound\_feature that need not be unique.

## 4.2.41 Compound\_feature\_element

The Compound\_feature\_element specifies the type of Machining\_feature (see 4.2.177) or Transition\_feature (see 4.2.335) to be used for a particular element of a Compound\_feature. A Compound\_feature consists of two or more Compound\_feature\_element objects and need not require Compound\_feature\_element objects to be ordered.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.40.



**Figure 7 — Compound\_feature**

The data associated with a `Compound_feature_element` are the following:

- element.

#### 4.2.41.1 element

The element specifies the base feature to be used as one of the components for the `Compound_feature`. See 4.3.60 and 4.3.61 for the application assertions.

#### 4.2.42 Compound\_feature\_relationship

A `Compound_feature_relationship` defines the sequence in which the `Compound_feature` elements are applied in the `Compound_feature`. The `Compound_feature_relationship` defines which feature is the preceding feature and which is the succeeding feature.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.41.

The data associated with a `Compound_feature_relationship` are the following:

- predecessor;
- successor.

#### **4.2.42.1 predecessor**

The predecessor specifies the `Compound_feature_element` with the highest precedence. See 4.3.62 for the application assertion.

#### **4.2.42.2 successor**

The successor specifies `Compound_feature_element` with a lesser precedence. See 4.3.62 for the application assertion.

### **4.2.43 Condition\_or\_assumption**

A `Condition_or_assumption` is a state of physical phenomena or presumption about the state of physical phenomena. The data associated with a `Condition_or_assumption` are the following:

- `associated_property`;
- `description`;
- `further_information`.

#### **4.2.43.1 associated\_property**

The `associated_properties` specifies the value of a physical property, or the functional dependence of a physical property, that represents the `Condition_or_assumption`. The associated properties need not be specified for a particular `Condition_or_assumption`. See 4.3.64 or 4.3.65 for the application assertion.

#### **4.2.43.2 description**

The `description` specifies the textual account of the `Condition_or_assumption`.

#### **4.2.43.3 further\_information**

The `further_information` specifies the additional document that describes or delineates the `Condition_or_assumption`. The further information need not be specified for a particular `Condition_or_assumption`. See 4.3.63 for the application assertion.

### **4.2.44 Connection\_transition**

A `Connection_transition` is the additional information defining fillet and edge round radius. The data associated with a `Connection_transition` are the following:

- `edge_round`;
- `fillet`.

#### 4.2.44.1 edge\_round

The `edge_round` specifies the radius that transitions the side of the channel to the top of the channel. See 4.3.66 for the application assertion.

#### 4.2.44.2 fillet

The `fillet` specifies the radius that transitions the ingate side to the parting surface between the production pattern and the runner. See 4.3.66 for the application assertion.

### 4.2.45 Continuous\_process

A `Continuous_process` organizes `Manufacturing_process` (see 4.2.180) in a sequential order for a specific machine with a specific machine setup of a process plan.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.16.

The data associated with a `Continuous_process` shall be one of the following:

- `process_type`;
- `related_process`;
- `relating_process`.

#### 4.2.45.1 process\_type

The `process_type` specifies the kind or kinds of continuous process is being done. The values of the `content_geometry_type` may be one of the following:

- `batch`;
- `serial`;
- `serial and batch`.

NOTE See 4.2.45.1.1 to 4.2.45.1.3 for the definition of each allowable value for `process_type`.

**4.2.45.1.1 batch:** the related process and the relating process are executed for each lot size.

**4.2.45.1.2 serial:** the related process and the relating process are executed for each part.

**4.2.45.1.3 serial and batch:** the related and relating process are both batch and serial.

### **4.2.45.2 related\_process**

The `related_process` specifies the sequence of process to be processed for a specific machine with a specific machine setup by the process plan. The sequencing is achieved through chaining together the `Manufacturing_process` objects (see 4.2.180). The `related_process` references the next `Manufacturing_process`. See 4.3.67 for the application assertion.

### **4.2.45.3 relating\_process**

The `relating_process` specifies the sequence of process to be processed for a specific machine with a specific machine setup by the process plan. The sequencing is achieved through chaining together the `Manufacturing_process` objects (see 4.2.180). The `relating_process` references the previous `Manufacturing_process`. See 4.3.67 for the application assertion.

## **4.2.46 Controller**

The Controller is the computerized numerical controller that give instructions to a machine for the purpose of creating a product.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.18.

The data associated with a Controller are the following:

- `controller_model`;
- `cocontroller_name`;
- `controller_specification`.

### **4.2.46.1 controller\_model**

The `controller_model` specifies the descriptive model identification of the controller. The `controller_model` need not be specified for a particular Controller.

### **4.2.46.2 controller\_name**

The `controller_name` specifies the descriptive name for the controller.

### **4.2.46.3 controller\_specification**

The `controller_specification` specifies the hardcopy or digital document that details information about the controller. See 4.3.68 for the application assertion.



## 4.2.47 Cooling\_port

A Cooling\_port is an inlet and outlet that permits a liquid or air to pass through a die mould to quickly reduce the mould temperature. The data associated with a Cooling\_port are the following:

- cooling\_size;
- male\_or\_female;
- port\_location.

### 4.2.47.1 cooling\_size

The cooling\_size specifies the dimensional parameters for cooling inlets and outlets for the production mould. See 4.3.69 for the application assertion.

### 4.2.47.2 male\_or\_female

The male\_or\_female specifies a Boolean value that indicates if the production mould cooling inlet and outlet port is male or female. A value of true specifies that cooling port is male. A value of false specifies that cooling port is female.

### 4.2.47.3 port\_location

The port\_location specifies the exact position or point for placing the cooling port in the mould. See 4.3.70 for the application assertion.

## 4.2.48 Core

A Core is a type of Sand\_casting\_design\_feature (see 4.2.276) that is is a design of specially formed material inserted into a mould to shape the interior or other part of a casting that cannot be easily shaped by the pattern.

NOTE 1 Figure 8 illustrates the Core with a Core\_print

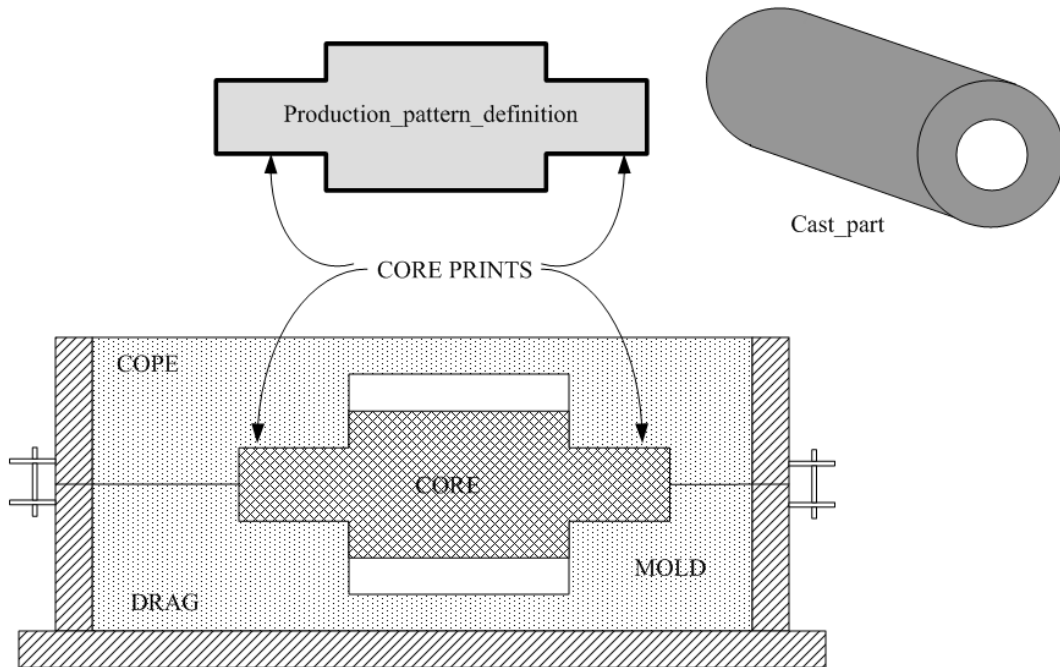
NOTE 2 This application object is harmonized with the core object from ISO 8062-1:2007, Clause 2.5.

The data associated with a Core are the following:

- core\_placement;
- core\_type.
- gas\_vent;
- material\_definition;

### 4.2.48.1 core\_placement

The `core_placement` specifies the sequential order for which cores are to be set in the mould when a placement sequence is required. A `core_placement` need not be specified for a particular Core. See 4.3.72 for the application assertion.



**Figure 8 — Core and Core\_print**

### 4.2.48.2 core\_type

The `core_type` specifies type of core being used.

NOTE See 4.2.48.2.1 to 4.2.48.2.8 for the definition of each allowable value for `core_type`.

**4.2.48.2.1 baked core:** a core that has been heated for a specified time and temperature to produce the desired physical properties.

**4.2.48.2.2 cover core:** a core that is set in place during the ramming of a mould to cover and complete a cavity partly formed by the withdrawal of a loose piece on the pattern

**4.2.48.2.3 isocure:** a core made by a coldbox core system comprised of a specific resin-coated sand and then cured by a vaporized amine catalyst.

**4.2.48.2.4 ram-up core:** a core that is set against the pattern or in a location in the pattern. The mould is rammed and when the pattern is drawn, the core remains in the mould.

**4.2.48.2.5 ring core:** a core used for shaping contours to the outside diameter of a casting.

**4.2.48.2.6 setup core:** a core print made of a dry sand core to provide a stronger bearing for a heavy core should a green sand print prove inadequate. A common practice in large green sand work.

**4.2.48.2.7 slab\_core:** a plain flat core.

**4.2.48.2.8 SO<sub>2</sub> core:** a core made using a SO<sub>2</sub> gas system with a resin-coated sand.

### 4.2.48.3 gas\_vent

The gas\_vent specifies the dimensional values of the vent for gas to escape, and locations where core vents are placed for gas to escape. A gas\_vent need not be specified for a particular Core. See 4.3.71 for the application assertion.

### 4.2.48.4 material\_definition

The material\_definition specifies the material the core is made of.

NOTE See 4.2.48.4.1 to 4.2.48.4.4 for the definition of each allowable material\_definition.

**4.2.48.4.1 ceramic:** a refractory and binder substance when fired to a high temperature becomes a rigid structure.

**4.2.48.4.2 oil\_bonded\_sand:** a sand bonded with such oils as linseed and binders.

**4.2.48.4.3 plaster:** known as plaster of paris; a semi\_hydrated form of calcium sulfate by sintering gypsum to 120-130 C.

**4.2.48.4.4 resin\_coated\_sand:** a sand the is coated with an adhesive substance insoluble in water and the coating is activated by a catalyst.

### 4.2.49 Core\_assembly

A Core\_assembly is when each core for the casting is placed in the proper core print set. The data associated with a Core\_assembly are the following:

- hard\_form;
- soft\_copy.

#### **4.2.49.1 hard\_form**

The `hard_form` specifies one or more fixed or rigid devices used to ensure core assembly and placement in mould. The `hard_form` need not be specified for a particular a `Core_assembly` for a cast part. See 4.3.73 for the application assertion.

#### **4.2.49.2 soft\_copy**

The `soft_copy` specifies one or more paper reports or digital report file used to ensure core assembly and placement in mould. The `soft_copy` need not be specified for a particular a `Core_assembly` for a cast part. See 4.3.74 for the application assertion.

#### **4.2.50 Core\_bench**

A `Core_bench` is a type of `Core_equipment` (see 4.2.54) that is a place where small cores are made, often referred to as bench cores. The data associated with a `Core_bench` are the following:

- `cores_required`;
- `wax_vent`.

##### **4.2.50.1 cores\_required**

The `cores_required` specifies the total number of cores to be produced as bench cores for a specific cast part.

##### **4.2.50.2 wax\_vent**

The `wax_vent` specifies the word or group of words that describes the type, size and location of the wax vent.

#### **4.2.51 Core\_box\_tooling**

A `core_box_tooling` is a type of `Tooling` (see 4.2.333) that is required to produce a core when a core is necessary to make a cast part. The data associated with a `Core_box_tooling` are the following:

- `core_box_equipment`.

##### **4.2.51.1 core\_box\_equipment**

The `core_box_equipment` specifies the core box requirements to produce a core. There may be more than one `core_box_equipment` for a particular `Core_box_tooling`. See 4.3.75 for the application assertion.

## 4.2.52 Core\_box\_vent

A Core\_box\_vent is a perforated object that is inserted into a core box to permit air to escape as core sand is forced into the core box under pressure. The data associated with a Core\_box\_vent are the following:

- core\_shape;
- material\_type;
- vent\_dimensions;
- vent\_type.

### 4.2.52.1 core\_shape

The core\_shape specifies the vent face shape to match the core face in the core box where the vent is to be inserted. See 4.3.77 for the application assertion.

NOTE sometimes the face of the vent is to be formed to the core box surface at the point of entry of the vent into the core box. The contour should be smooth and uniform with the core box surface.

### 4.2.52.2 material\_type

The material\_type specifies the vent substance.

EXAMPLE Brass, stainless steel and nylon are examples of vents.

The values of the material\_type may be one of the following:

- brass;
- nylon;
- polyurethane;
- stainless\_steel.

NOTE See 4.2.52.2.1 to 4.2.52.2.4 for the definition of each allowable value for material\_type.

**4.2.52.2.1 brass:** vent body is made of solid brass.

**4.2.52.2.2 nylon:** vent body is made of solid nylon.

**4.2.52.2.3 polyurethane:** vent body is made of polyurethane.

**4.2.52.2.4 stainless\_steel:** vent is made of solid stainless steel.

### **4.2.52.3 vent\_dimensions**

The `vent_dimensions` specifies the dimensional form of the vent placed in the core box. See 4.3.76 for the application assertion.

### **4.2.52.4 vent\_type**

The `vent_type` specifies style of each vent used in the core box.

EXAMPLE Styles of vents may be slotted, screen, vibratory, etc.

## **4.2.53 Core\_conveyer\_system**

A `Core_conveyer_system` is a type of `Core_equipment` (see 4.2.54) that is used with the production of no bake or air set cores. The conveyer cycles the core boxes into the mixer and sand dispenser for filling the box then to the core box stripping area and back to the mixer and sand dispenser for filling. The data associated with a `Core_conveyer_system` are the following:

- `cores_required`;
- `core_box_skid`;
- `core_plug`;
- `cycle_time`.

### **4.2.53.1 cores\_required**

The `cores_required` specifies the total number of cores to be produced with the conveyer system for a specific cast part.

### **4.2.53.2 core\_box\_skid**

The `core_box_skid` specifies the size of the skid plate or board to move easily on the conveyer system. See 4.3.78 for the application assertion.

### **4.2.53.3 core\_plug**

The `core_plug` specifies the wedge block size for the cavity that will be filled with dry sand to aid in core sand removal after casting cools. See 4.3.78 for the application assertion.

### **4.2.53.4 cycle\_time**

The `cycle_time` specifies the time a core shall remain in the core box. Calculated from time of fill to time of stripping the core from the core box.

## 4.2.54 Core\_equipment

A Core\_equipment is a type of Equipment (see 4.2.110) that is used to produce cores or for core handling. Each Core\_equipment is either a Core\_machine (see 4.2.55), Core\_bench (see 4.2.50), Core\_conveyer\_system (see 4.2.53), Core\_setter (see 4.2.58), Dryer (see 4.2.99) or Bankout\_frame (see 4.2.14).

## 4.2.55 Core\_machine

A Core\_machine is a type of Core\_equipment (see 4.2.54) that is a machine that will force sand through an invest opening(s) to fill the core box cavity with core sand. The data associated with a Core\_machine are the following:

- ejector\_definition;
- tool\_mounting\_definition;
- vent\_definition.

### 4.2.55.1 ejector\_definition

The ejector\_definition defines the ejector plate size and associated pins required for the Ejector\_system. See 4.3.79 for the application assertion.

### 4.2.55.2 tool\_mounting\_definition

The tool\_mounting\_definition specifies the requirements for attaching the core box to a core making machine. See 4.3.80 for the application assertion.

### 4.2.55.3 vent\_definition

The vent\_definition specifies the type of vent, location, and size of the vent installed in the core box. See 4.3.81 for the application assertion.

## 4.2.56 Core\_master

A Core\_master is a type of Core\_box\_tooling (see 4.2.51) that may be constructed to create the Production\_core\_box.

NOTE 1 A master core could be a core blank used as a tracing model for the production core box

NOTE 2 The master core could be a replica of the production core box built with double shrinkage and cast to size of the production core box

The data associated with a Core\_master are the following:

- core\_box\_master\_shrink\_factor;

### **4.2.56.1 core\_box\_master\_shrink\_factor**

The `core_box_master_shrink_factor` specifies the patternmaker's rule that compensates for double shrinkage. One shrinkage is required the casting shrinking in size as it cools to ambient temperature and the other shrinkage is required for the production tooling as it comes to ambient temperature.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE 1 if the cast part has a shrink factor of 1/8 inches and the production tooling has a shrink factor of 1/8 inches the master shrink factor is 1/4 inches or double shrinkage.

EXAMPLE 2 if the shrinkage compensation is one quarter inch per foot the shrink rule will be one quarter of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0208333 inches per inch or where the foot shrink rule will be 12.25 longer than the standard 12 inch foot rule.

### **4.2.57 Core\_print**

The `Core_print` is a type of `Sand_casting_design_feature` (see 4.2.276) that is attached to the `core_box` designed to form a seat to locate and support a core within the mould.

NOTE 1 When a core print is attached within a core box there will be at least two `Core_box(es)` required for the `Part_version`.

NOTE 2 Figure 8 illustrates the `Core` with a `Core_print`

NOTE 3 This application object is harmonized with the core print object from ISO 8062-1:2007, Clause 2.7.

The data associated with a `Core_print` are the following:

- `core_print_dimensions`;
- `clearance`;
- `draft`
- `feature`.

#### **4.2.57.1 core\_print\_dimensions**

The `core_print_dimensions` specifies the dimensional size of the seat form. See 4.3.84 for the application assertion.

#### **4.2.57.2 clearance**

The `clearance` specifies the differential distance between the core seat made by the core print and the core. See 4.3.84 for the application assertion.

NOTE the cavity made by the core print is to be larger than the core form to set into that cavity.



### 4.2.57.3 draft

The draft specifies an angle in degrees of taper assigned to each vertical core print surface that is not parallel with the Core\_print parting line. See 4.3.82 for the application assertion.

### 4.2.57.4 feature

The feature specifies the geometric surfaces that form the core print. See 4.3.83 for the application assertion.

## 4.2.58 Core\_setter

A Core\_setter is a type of Core\_equipment (see 4.2.54) that is used with a specific moulding machine to set or place cores in the mould. The data associated with a Core\_setter are the following:

- moulding\_machine.

### 4.2.58.1 moulding\_machine

The moulding\_machine specifies the word or group of words that describes the moulding machine with a core setter.

## 4.2.59 Corrosion\_service

The Corrosion\_service is the type of environmental or material conditions an in-service application will apply to the cast part. The data associated with a Corrosion\_service t are the following:

- corrosive\_environment;
- corrosive\_material;
- ph\_value;
- temperature.

### 4.2.59.1 corrosive\_environment

The corrosive\_environment specifies an atmospheric or ambient condition that has a erosive or acidic reaction to the cast part as designed. The corrosive\_environment need not be specified for a particular Corrosion\_service for the cast part. See 4.3.85 for the application assertion.

### 4.2.59.2 corrosive\_material

The corrosive\_material specifies a substance that has a erosive or acidic reaction to the cast part as designed. The corrosive\_material need not be specified for a particular Corrosion\_service for the cast part. See 4.3.85 for the application assertion.

### **4.2.59.3 ph\_value**

The `ph_value` specifies the measure of its acidity or alkalinity with units for which the cast part has been designed. The `ph_value` need not be specified for a particular `Corrosion_service` for the cast part. See 4.3.86 for the application assertion.

### **4.2.59.4 temperature**

The `temperature` specifies an operating temperature with units that causes a corrosive reaction to the cast part as designed. The `temperature` need not be specified for a particular `Corrosion_service` for the cast part. See 4.3.86 for the application assertion.

### **4.2.60 Curve\_point**

A `Curve_point` is location of a data point on a `Data_curve` (see 4.2.67). The data associated with a `Curve_point` are the following:

- `x_value`
- `y_value`

#### **4.2.60.1 x\_value**

The `x_value` specifies the numerical measure of the x-coordinate for the `Curve_point`.

#### **4.2.60.2 y\_value**

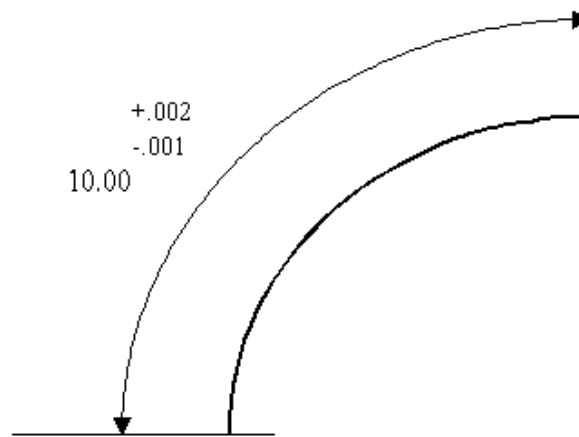
The `y_value` specifies the numerical measure of the y-coordinate for the `Curve_point`.

### **4.2.61 Curved\_dimension\_tolerance**

A `Curved_dimension_tolerance` is a type of `Size_tolerance` (see 4.2.304) that is the tolerance on a dimension for a curve measured along the entire path of the curve.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.48.

NOTE 2 Figure 9 illustrates the `Curved_dimension_tolerance`.



**Figure 9 — Curved dimension tolerance**

#### 4.2.62 Customer\_casting\_requirement

The Customer\_casting\_requirements specify the necessary conditions that are to be executed or achieved for the production of the cast part. Each Customer\_casting\_requirements is either a Casting\_service\_data (see 4.2.27), Inspection\_or\_test\_requirement, (see 4.2.154), Pattern\_equipment\_available (see 4.2.220), Payment\_and\_shipping (see 4.2.223), Process\_requirement (see 4.2.240), Reporting\_requirement (see 4.2.262), Special\_inspection\_requirement (see 4.2.306), or Tolerance\_requirement (see 4.2.328).

NOTE the conditions are determined either by the customer, the foundry, or the machine shop.

The data associated with a Customer\_casting\_requirements are the following:

- applies\_to
- description;
- new\_tooling\_required;
- illustration
- referenced\_from;
- specific\_comments.

#### **4.2.62.1 applies\_to**

An `applies_to` specifies a precise requirement to a particular shape of the part to be cast. See 4.3.88 for the application assertion.

#### **4.2.62.2 description**

The description summarizes with words the specific requirement for the cast part.

#### **4.2.62.3 illustration**

An illustration specifies a drawing or sketch in a graphical form for a particular aspect of the part to which a requirement for the cast part is applied. See 4.3.87 for the application assertion.

#### **4.2.62.4 new\_tooling\_required**

The `new_tooling_required` specifies that tooling shall be produced before a cast part can be cast for the selected moulding process. The `new_tooling_required` need not be specified for a particular a `Customer_casting_requirement` for the cast part.

NOTE See 4.2.62.4.1 to 4.2.62.4.2 for the definition of each allowable value for `new_tooling_required`

**4.2.62.4.1 Customers Tooling Shop :** The pattern or die tooling shop used by the customer for their parts that require the casting process.

**4.2.62.4.2 Foundry Tooling Shop :** The pattern or die tooling shop used by the foundry for their cast parts.

#### **4.2.62.5 referenced\_from**

The `referenced_from` specifies the particular document for which the specification is called out for the requirement. See 4.3.89 for the application assertion.

#### **4.2.62.6 specific\_comments**

The `specific_comment` specifies special instructions related to the cast part. There may be more than one `specific_comment` for a `Customer_casting_requirement` for a cast part. The `specific_comment` need not be specified for a particular a `Customer_casting_requirement` for the cast part.

### 4.2.63 Customer\_design\_deficiency\_report

A Customer\_design\_deficiency\_report is a type of Document\_assignment (see 4.2.94) that outlines flaws, defects, or other problem areas in the part design that will cause imperfection, weaknesses, failures in the cast part. The data associated with an Customer\_design\_deficiency\_report are the following:

- list\_of\_results.

#### 4.2.63.1 list\_of\_results

A list\_of\_results is a documentation of simulation findings.

### 4.2.64 Customer\_order

A Customer\_order is a document, received from a customer, that describes data to process a request for the manufacture of a cast part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.49.

The data associated with a Customer\_order are the following:

- casting\_method;
- customer;
- delivery\_date;
- first\_article\_requirement;
- initiated\_order;
- inspection\_procedure;
- material\_disposition;
- order\_number;
- order\_status;
- production\_status;
- quantity\_ordered;

ISO 10303-223:2008 (E)

- rfq;
- simulation\_run\_result;
- special\_instructions.

#### **4.2.64.1 casting\_method**

The casting\_method specifies the moulding process to be used for the cast part. See 4.2.204.3 for non permanent moulding process and 4.2.226.4 for permanent moulding process.

#### **4.2.64.2 customer**

The customer specifies the person and company requesting the manufacture of the part. See 4.3.94 for the application assertion.

#### **4.2.64.3 delivery\_date**

The delivery\_date specifies the year, month and day when the part is to be received by the customer. See 4.3.90 for the application assertion.

#### **4.2.64.4 first\_article\_requirement**

The first\_article\_requirement specifies the requirements for inspection and approval for full production of the cast part requested. See 4.3.91 for the application assertion.

#### **4.2.64.5 initiated\_order**

The initiated\_order specifies Project\_order that is initiated by the Customer\_order. There may be more than one initiated\_order for a Customer\_order. See 4.3.95 for the application assertion.

#### **4.2.64.6 inspection\_procedure**

The inspection\_procedure specifies the tactics and rules by which the Inspection\_plan shall be conducted and applied. There may be more than one inspection\_procedure for a Customer\_order requested for a cast part. See 4.3.92 for the application assertion.

#### **4.2.64.7 material\_disposition**

The material\_disposition specifies the instructions for handling of material that has been delivered to manufacture a part.

#### **4.2.64.8 order\_number**

The order\_number specifies a unique identification of the Customer\_order.

**4.2.64.9 order\_status**

The `order_status` specifies the disposition of the `Customer_order`. The disposition need not be unique.

**4.2.64.10 production\_status**

The `production_status` specifies the stage for which the cast part requested is in production.

NOTE See 4.2.64.10.1 to 4.2.64.10.4 for the definition of each allowable value for status.

**4.2.64.10.1 current production:** The part to be cast is being produced currently at another foundry.

**4.2.64.10.2 new production:** The part to be cast has never be produced by a foundry as a cast part.

**4.2.64.10.3 experimental:** The part to be cast is a research and development part.

**4.2.64.10.4 prototype:** The part is at some stage of a research and development design but not at new production status.

**4.2.64.11 quantity\_ordered**

The `quantity_ordered` specifies number of parts to be manufactured (see `Ordered_part` 4.2.209). There may be more than one `quantity_ordered` for a `Customer_order`. See 4.3.93 for the application assertion.

**4.2.64.12 rfq**

The RFQ specifies a quotation request for the purchase of a cast part or tooling to produce a cast part. There may be more than one RFQ for a `Customer_order`. The RFQ need not be specified for a particular `Customer_order`. See 4.3.96 for the application assertion.

**4.2.64.13 simulation\_run\_result**

The `simulation_run_result` specifies the conclusions of a simulation study or analysis. The `simulation_run_result` need not be specified for a particular a `Customer_order` for a cast part. See 4.3.97 for the application assertion.

**4.2.64.14 special\_instructions**

The `special_instructions` specifies directions for processing of a `Customer_order` that apply to manufacturing a part which are not common and need to be described and documented.

## **4.2.65 Customer\_simulation**

The `Customer_simulation` is a type of `Simulation_process` (see 4.2.295) that request from a customer to use a simulation software tool to produce a simulation of a casting process for a proposed cast part.

The data associated with an `Customer_simulation` are the following:

- `defect_results`;
- `suggested_tool_design`.

### **4.2.65.1 defect\_results**

The `defect_results` specifies a list of one or more part design defects identified by the simulation software run and the data analysis from the simulation run. See 4.3.98 for the application assertion.

### **4.2.65.2 suggested\_tool\_design\_report**

The `suggestion_tool_design_report` specifies an optimum tooling design for producing a quality casting for the part design as identified by the simulation software run and the data analysis from the simulation run. See 4.3.99 for the application assertion.

## **4.2.66 Customer\_tool\_design\_report**

A `Customer_tool_design_report` is a type of `Document_assignment` (see 4.2.94) that outlines changes to the part design and recommendation for a gating system that would meet production rates and produce a quality cast part. The data associated with an `Customer_customer_tool_design_report` are the following:

- `designed_tool`.

### **4.2.66.1 designed\_tool**

The `designed_tool` specifies the tooling design shape required to satisfy the customer requirements for the cast part specified. See 4.3.99 for the application assertion.

## **4.2.67 Data\_curve**

A `Data_curve` provides a set of data points for analyzing measurement values to determine expect results.



The data associated with a Data\_curve are the following:

- interpolation\_method;
- x\_unit;
- y\_unit;
- data\_points.

#### **4.2.67.1 interpolation\_method**

The interpolation method specifies the procedure for estimation of property values in between adjoining sets of data points.

NOTE Interpolation methods include linear and polynomial interpolation.

#### **4.2.67.2 x\_unit**

The x unit specifies the unit of measure for the ordinate of the Data curve.

#### **4.2.67.3 y\_unit**

The y unit specifies the unit of measure for the abscissa of the Data curve.

#### **4.2.67.4 data\_points**

The data point specifies the pair of values for two variables through which the Data curve shall pass. There may be more than one data point for a Data curve. See 4.3.100 for the application assertion.

#### **4.2.68 Date**

The Date defines a unique day, month, and year as an alphanumeric string or as a number.

#### **4.2.69 Datum**

A Datum is defined in ISO 5459. A Datum is either a Datum\_feature or a Compound\_-datum.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.54.

The data associated with a Datum are the following:

- name;
- precedence.

### **4.2.69.1 name**

The name specifies a word by which a Datum is commonly called. This name shall be unique within a part.

### **4.2.69.2 precedence**

The precedence specifies a sequence that datums are applied when there are two or more datums required for the definition of a Geometric\_tolerance.

## **4.2.70 Datum\_feature**

A Datum\_feature is a type of Datum (see 4.2.69) that is a feature on a part used to establish a Datum.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.55.

The data associated with a Datum\_feature are the following:

- datum\_representation;
- modifier.

### **4.2.70.1 datum\_representation**

The datum\_representation specifies the type of representation for defining a Datum\_feature. The Datum\_feature may be defined by either a Datum\_target definition or by Part shape representation. See 4.3.102 and 4.3.103 for the application assertions.

### **4.2.70.2 modifier**

The modifier specifies the tolerance value applied to the Datum\_feature. The modifier need not be specified for a particular Datum\_feature. See 4.3.101 for the application assertion.

## **4.2.71 Datum\_target**

A Datum\_target is a geometric element on the surface of a part to locate a Datum for reference by a Geometric\_tolerance. A Datum\_target is either a Placed\_target (see 4.2.231) or a Target\_area (see 4.2.318).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.56.

The data associated with a Datum\_target are the following:

- identifier.

#### **4.2.71.1 identifier**

The identifier specifies a unique identification for the Datum\_target.

#### **4.2.72 Datum\_target\_set**

A Datum\_target\_set is a set of Datum\_target objects that are used to define a datum reference for a Geometric\_tolerance.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.57.

The data associated with a Datum\_target\_set are the following:

- rule\_description;
- target\_shape.

##### **4.2.72.1 rule\_description**

The rule\_description specifies the type of datum that is formed by the Datum\_target\_set. The rule\_description need not be specified for a particular Datum\_target\_set.

EXAMPLE "V-block" indicates that two Datum\_target objects on a cylindrical element are to form the areas of contact on a V-shaped fixture.

##### **4.2.72.2 target\_shape**

The target\_shape specifies the set of Datum targets that define the Datum\_target\_set. There may be more than one target\_shape for a Datum\_target. See 4.3.104 for the application assertion.

#### **4.2.73 Defect\_prediction**

A Defect\_prediction is an undesirable structural feature that is forecast by a simulation.

EXAMPLE Possible defects include shrinkage porosity, oxide inclusions, and cracks.

ISO 10303-223:2008 (E)

The data associated with a Defect\_prediction are the following:

- defect\_shape;
- description;
- location;
- type\_of\_defect.

#### **4.2.73.1 defect\_shape**

The defect\_shape specifies the characteristic spatial form of a defect. The defect\_shape need not be specified for a particular Defect\_prediction. See 4.3.105 for the application assertion.

#### **4.2.73.2 description**

The description specifies the textual account of a Defect prediction.

#### **4.2.73.3 location**

The location specifies the position of a predicted defect within a part. The location need not be specified for a particular Defect prediction. See 4.3.106 for the application assertion.

#### **4.2.73.4 type\_of\_defect**

The type\_of\_defect specifies the label that identifies the kind of defect.

NOTE Shrinkage, porosity, cracks and oxide inclusions are kinds of defects.

#### **4.2.74 Descriptive\_parameter**

A Descriptive\_parameter is a type of Property\_parameter (see 4.2.256) that is an explanation of the property being defined by a specification.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.62.

The data associated with a Descriptive\_parameter are the following:

- descriptive\_string.

#### **4.2.74.1 descriptive\_string**

The descriptive\_string specifies a word or group of words by which a Descriptive\_parameter is explained.

## 4.2.75 Design\_exception\_notice

A `Design_exception_notice` is a notification of a design discrepancy discovered during the creation of the process plan for a given part such that process planning cannot continue until a technical recommendation is made to correct the problem.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.63.

The data associated with a `Design_exception_notice` are the following:

- `discrepant_part`;
- `issues`;
- `issuing_date`;
- `notice_description`;
- `notice_number`;
- `technical_recommendation`.

### 4.2.75.1 discrepant\_part

The `discrepant_part` specifies the part that has a design discrepancy. There may be more than one `discrepant_part` for a `Design_exception_notice`. See 4.3.109 for the application assertion.

### 4.2.75.2 issues

The `issues` specifies the change proposal to modify a part. The `issues` need not be specified for a particular `Design_exception_notice`. There may be more than one `issues` for a `Design_exception_notice`. See 4.3.108 for the application assertion.

### 4.2.75.3 issuing\_date

The `issuing_date` specifies the year, month and day when the `design_exception_notice` was created. See 4.3.107 for the application assertion.

### 4.2.75.4 notice\_description

The `notice_description` specifies the kind of problem or non-conformance machining condition causing a rejection of a part.

EXAMPLE If two holes were drilled simultaneously and the drill bits would run together, a recommendation would be needed to either change the depth of the hole or change the machining process.

#### **4.2.75.5 notice\_number**

The `notice_number` specifies a unique identification for each `Design_exception_notice`.

#### **4.2.75.6 technical\_recommendation**

The `technical_recommendation` specifies a recommended resolution to a design problem discovered during the creation of the process plan for a part. The `technical_recommendation` need not be specified for a particular `Design_exception_notice`.

#### **4.2.76 Design\_part**

A `Design_part` is a type of `Part_version` (see 4.2.217) that is the engineering design specification for a part to be manufactured as originally received from the client.

NOTE The customer may not be aware of, or be concerned with, the limitations of the casting process. Therefore, a `Design_part` may need to be modified to enable it to be economically produced by casting.

The data associated with a `Design_part` are the following:

— `design_feature`;

##### **4.2.76.1 design\_feature**

The `design_feature` specifies a feature that is required to be manufactured or is to be included in the part design. There may be more than one `design_feature` for a `Design_part`. The `design_feature` need not be specified for a particular `Design_part`. See 4.3.110 for the application assertion.

#### **4.2.77 Design\_reference**

A `Design_reference` is the identification of a graphic representation used to communicate the design. It identifies a version of a document that depicts the physical and functional data for a product, or some characteristic thereof, by means of pictorial and text presentations.

NOTE 1 This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.24.

NOTE 2 A `Design_reference` may be of type DXF, CAD, or other representation type.

The data associated with a `Design_reference` are the following:

— `drawing_data_file`;

— `drawing_type`;

— `document_version`;

- id;
- identifies;
- revision\_level.

#### **4.2.77.1 drawing\_data\_file**

The `drawing_data_file` specifies the hardcopy or digital document that is the drawing. See 4.3.111 for the application assertion.

#### **4.2.77.2 drawing\_type**

The `drawing_type` specifies the word, or group of words, that identify the item within the category of Drawings.

EXAMPLE A `drawing_type` may include such word groups as a tool design drawing, an engineering drawing, or a manufacturing drawing.

#### **4.2.77.3 document\_version**

The `documents_version` specifies the design version of the product referenced for a `Part_version`. `Design_reference` for the A `documents_version` need not be specified for a particular `Part_version`. See 4.3.113 for the application assertion.

EXAMPLE a `documents` level could be a Preliminary design, Approved for Production or others.

#### **4.2.77.4 id**

The `id` specifies the unique identifier for a `Design_reference` within an organization.

#### **4.2.77.5 identifies**

The `identifies` specifies the activity that requires a `Design_reference` for additional information. The `identifies` need not be specified for a particular `Design_reference`. See 4.3.112 for the application assertion.

#### **4.2.77.6 revision\_level**

The `revision_level` specifies the `Design_reference` release level. A `Design_reference` release level corresponds to a particular product version reflecting an approved engineering change.

#### 4.2.78 Diameter\_dimension\_tolerance

A Diameter\_dimension\_tolerance is a type of Size\_tolerance (see 4.2.304) that is the allowable variation of the size of a hole in a surface.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.65.

NOTE 2 Figure 10 illustrates the Diameter\_dimension\_tolerance.

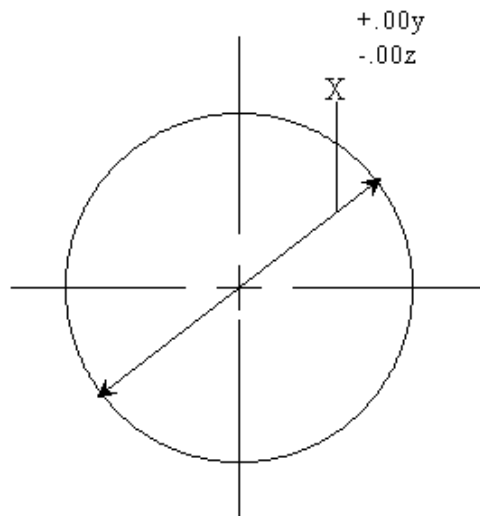


Figure 10 — Diameter\_dimension\_tolerance

#### 4.2.79 Die\_cast\_master

A Die\_cast\_master is a type of Die\_cast\_tooling (see 4.2.80) that is a scale replica of the production tooling needed to produce cast part. The scale factor or shrink multiplier will be determined by the material and the process used to produce the production tooling.

The data associated with a Die\_cast\_master are the following:

- female\_component;
- male\_component;
- machining\_allowance\_on\_master\_tooling;
- master\_shrink\_factor.



#### 4.2.79.1 female\_component

The female\_component specifies the female tool from which the production tool is made. The female\_component need not be specified for a particular Die\_cast\_master. See 4.3.114 for the application assertion.

#### 4.2.79.2 male\_component

The male\_component specifies male tool from which the production tool is made. The male\_component need not be specified for a particular Die\_cast\_master. See 4.3.114 for the application assertion.

#### 4.2.79.3 machining\_allowance\_on\_master\_tooling

The machining\_allowance\_on\_master\_tooling specifies the stock to be added to the tool to permit machining. There may be more than one machining\_allowance\_on\_master\_tooling specified for a Die\_cast\_master. See 4.3.115 for the application assertion.

#### 4.2.79.4 master\_shrink\_factor

The master\_shrink\_factor specifies the patternmaker's rule that compensates for double shrinkage. One shrinkage is required the casting shrinking in size as it cools to ambient temperature and the other shrinkage is required for the production tooling as it comes to ambient temperature.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE 1 if the cast part has a shrink factor of 1/8 inches and the production tooling has a shrink factor of 1/8 inches the master shrink factor is 1/4 inches or double shrinkage.

EXAMPLE 2 if the shrinkage compensation is one quarter inch per foot the shrink rule will be one quarter of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0208333 inches per inch or where the foot shrink rule will be 12.25 longer than the standard 12 inch foot rule.

### 4.2.80 Die\_cast\_tooling

The Die\_cast\_tooling is a type of Tooling (see 4.2.333) that are the metal die components that form a reusable mould to produce a near\_net shape precision cast part. The data associated with a Die\_cast\_tooling are the following:

- die\_equipment;
- material\_definition.
- tooling\_identifier;
- tooling\_location.

#### **4.2.80.1 die\_equipment**

The die\_equipment specifies the information for the male die and the female die for a die cast mould. See 4.3.117 for the application assertion.

#### **4.2.80.2 material\_definition**

The material\_definition specifies primary material choice of raw stock for producing the Part. There may be more than one material\_definition for a part. See 4.3.128 for the application assertion.

#### **4.2.80.3 tooling\_identifier**

The tooling\_identifier specifies an alphanumeric string that identifies the tooling used to produce the cast part

#### **4.2.80.4 tooling\_location**

The tooling\_location specifies where a tool is stored. See 4.3.118 and 4.3.119 for the application assertion.

### **4.2.81 Die\_clamping**

A Die\_clamping is a type of die\_design\_feature (see 4.2.83) that is the mechanism that secures the die mould to the die casting machine. The data associated with a Die\_clamping are the following:

— die\_clamping\_positions.

#### **4.2.81.1 die\_clamping\_positions**

The die\_clamping\_position specifies the placement, direction, or point of attachment for securing the clamping mechanism for the die mould in the die casting machine. There may be one or more clamping positions for securing the die mould to the die casting machine. See 4.3.120 for the application assertion.

### **4.2.82 Die\_core**

A Die\_core is a design of specially formed material inserted into a die mould to shape the interior or other part of a casting that cannot be easily shaped or formed by the die mould. The data associated with a Die\_core are the following:

— core\_type;

— draft;

— guide\_pin;

— metal\_type;

— parting;

— shape.

### 4.2.82.1 core\_type

The core\_type specifies the type of core required

NOTE See 4.2.82.1.1 to 4.2.82.1.7 for the definition of each allowable value for core\_type.

**4.2.82.1.1 fixed\_core:** a metal core that is attached to the die mould.

**4.2.82.1.2 ceramic:** a core made with a refractory and binder material then fired at a high temperature.

**4.2.82.1.3 mechanical\_slide:** a movable metal core that is position in and retracted from the die mould mechanically.

**4.2.82.1.4 hydraulic\_slide:** a movable metal core that is position in and retracted from the die mould hydraulically.

**4.2.82.1.5 pneumatic slide:** a movable metal core that is position in and retracted from the die mould pneumatically.

**4.2.82.1.6 salt\_core:** a core made of salt and binder material that withstand molten metal temperatures.

**4.2.82.1.7 sand\_core:** a core made of sand and a binder material.

### 4.2.82.2 draft

The draft specifies an angle in degrees of slope or taper. See 4.3.121 for the application assertion.

### 4.2.82.3 guide\_pin

The guide\_pin specifies one or more pin diameters and length for the pins used to located the Die\_core. The guide\_pin need not be specified for a particular Die\_core. See 4.3.122 for the application assertion.

### 4.2.82.4 metal\_type

The metal\_type specifies the metal alloy used for or the plating used on the Die\_core.

### **4.2.82.5 parting**

The parting specifies the surfaces that define the parting for the Die\_core used in the production mould. See 4.3.123 for the application assertion.

### **4.2.82.6 shape**

The shape specifies the geometrical shape and form with the dimensional values of the core. See 4.3.124 for the application assertion.

### **4.2.83 Die\_design\_feature**

The Die\_design\_feature is a type of Feature (see 4.2.124) that is required in the design of a die cast mould. Each Die\_design\_feature is either a Cooling\_port (see 4.2.47), a Die\_clamping (see 4.2.81), a Die\_mould\_vent (see 4.2.87), a Secondary\_tooling (see 4.2.279), a Shot\_sleeve (see 4.2.288), or a Slide (see 4.2.305). The data associated with a Die\_design\_feature are the following:

— placement;

#### **4.2.83.1 placement**

The placement specifies the position and orientation of a Die\_design\_feature relative to the base shape of a die for the part. 4.3.125 See for the application assertion.

### **4.2.84 Die\_master**

A Die\_master is a type of die\_mould\_tooling (see 4.2.86) that is a scaled replica of the production die needed to produce a cast part. The scale factor will be determined by the material and the process used to produce the Production\_die\_mould.

The data associated with a Die\_master are the following:

— female\_component;

— machining\_allowance\_on\_tooling;

— male\_componen;

— master\_shrink\_factor.

#### **4.2.84.1 female\_component**

The female\_component specifies female tool from which the production tool is cast. See 4.3.126 for the application assertion.

#### 4.2.84.2 male\_component

The male\_component specifies male tool from which the production tool is cast. See 4.3.126 for the application assertion.

#### 4.2.84.3 machining\_allowance\_on\_tooling

The machining\_allowance\_on\_tooling specifies the stock to be added to the tool to permit machining. There may be more than one machining\_allowance\_on\_tooling specified for a Die\_master. See 4.3.127 for the application assertion.

NOTE When a Production die is cast to near net shape, machine stock is added to the parting surface(s) and removable components for precision register, matches, and fits.

#### 4.2.84.4 master\_shrink\_factor

The master\_shrink\_factor specifies the patternmaker's rule that compensates for double shrinkage. One shrinkage is required the casting shrinking in size as it cools to ambient temperature and the other shrinkage is required for the production tooling as it comes to ambient temperature.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE 1 if the cast part has a shrink factor of 1/16 inches and the production tooling has a shrink factor of 3/16 inches the master shrink factor is 1/4 inches or double shrinkage.

EXAMPLE 2 if the shrinkage compensation is one quarter inch per foot the shrink rule will be one quarter of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0208333 inches per inch or where the foot shrink rule will be 12.25 longer than the standard 12 inch foot rule.

#### 4.2.85 Die\_mould\_requisition

A Die\_mould\_requisition is a type of Requisition (see 4.2.265) that describes dies required to support part manufacturing.

#### 4.2.86 Die\_mould\_tooling

A Die\_mould\_tooling is a type of Tooling (see 4.2.333) that is the die components that form a reusable mould to produce a near-net shape precision cast part.

ISO 10303-223:2008 (E)

The data associated with a `Die_mould_tooling` are the following:

- `die_equipment`;
- `material_definition`.
- `tooling_identifier`;
- `tooling_location`.

#### **4.2.86.1 die\_equipment**

The `die_equipment` specifies the male die and the female die for the mould. See 4.3.131 for the application assertion.

#### **4.2.86.2 material\_definition**

The `material_definition` specifies primary material choice of raw stock for producing the Part. There may be more than one `material_definition` for a part. See 4.3.128 for the application assertion.

#### **4.2.86.3 tooling\_identifier**

The `tooling_identifier` specifies an alphanumeric string that identifies the tooling used to produce the cast part

#### **4.2.86.4 tooling\_location**

A `tooling_location` specifies where a tool is stored. See 4.3.129 and 4.3.130 for the application assertion.

#### **4.2.87 Die\_mould\_vent**

A `Die_mould_vent` is a type of `die_design_feature` (see 4.2.83) that is a method by which trapped air can escape from a die cavity. The data associated with a `Die_mould_vent` are the following:

- `ejector_pin_definition`.

#### **4.2.87.1 ejector\_pin\_definition**

An `ejector_pin_definition` specifies the requirements for each pin used in the ejector system assembly as a vent for trapped air removal from a die cavity. See 4.3.132 for the application assertion.

#### **4.2.88 Digital\_file**

A `Digital_file` is a type of `Document_file` (see 4.2.95) that contains computer interpretable data.

## 4.2.89 Digital\_technical\_data\_package\_work\_order

A `Digital_technical_data_package_work_order` is a document that defines the need to create and track the digital product data definition for a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.68.

The data associated with a `Digital_technical_data_package_work_order` are the following:

— `order_id`.

### 4.2.89.1 order\_id

The `order_id` specifies a unique identifier for the `Digital_technical_data_package_work_order`.

## 4.2.90 Dimensional\_measurement\_inspection

A `Dimensional_measurement_inspection` is a type of `Inspection_activity` (see 4.2.153) that is a CMM procedure for evaluating the dimensional accuracy of a cast part against the design specification for every dimension with is associated tolerance for the part. The `Dimensional_measurement_inspection` may also be used to evaluate the dimensional accuracy of the tooling for a cast part.

The data associated with a `Dimensional_measurement_inspection` are the following:

— `iso10303_219_data`.

### 4.2.90.1 iso10303\_219\_data

The `ISO10303_219_data` specifies a reference to an external ISO standard that contains information defining the inspection requirements for inspecting parts that are produced by a manufacturing activity. The ISO 10303 219 is part of the suite of ISO 10303 manufacturing Application Protocols. See 4.3.133 for the application assertion.

NOTE ISO 10303 219:2007 is a standard for CMM inspection of a part and captures the Dimensional Measuring Interface Standard (DMIS) data from a CMM inspection activity.

### 4.2.91 Dimensional\_tolerance

A `Dimensional_tolerance` is the total amount a specific dimension is permitted to vary, which is the difference between maximum and minimum permitted limits of size. Each `Dimensional_tolerance` is either a `Location_tolerance` (see 4.2.169) or a `Size_tolerance` (see 4.2.304)

NOTE This application object is harmonized with the same object from 10303-224:2006, Clause 4.2.69.

EXAMPLE A dimension given as 1.624 +.002/-.002 means it may be 1.626 or 1.622, or anywhere between these limit dimensions.

The data associated with a Dimensional\_tolerance are the following:

- dimension\_description;
- dimension\_note;
- dimension\_value;
- limit;
- placement;
- significant\_digits;
- unit\_of\_measure.

#### **4.2.91.1 dimension\_description**

The dimension\_description specifies a textual description of any conditions which may affect the interpretation of the tolerance information that is defined. There may be more than one dimension\_description for a Dimensional\_tolerance. The dimension\_description need not be specified for a particular Dimensional\_tolerance.

EXAMPLE A Dimension\_tolerance may apply in 2 places.

#### **4.2.91.2 dimension\_note**

The dimension\_note specifies a qualifying note. There may be more than one dimension\_note for a Dimensional\_tolerance. The dimension\_note need not be specified for a particular Dimensional\_tolerance.

The values of the dimension\_note may be one of the following:

- auxiliary;
- theoretical;
- user defined.

NOTE See 4.2.91.2.1- 4.2.91.2.3 for the definition of each allowable value for dimensional\_note.

**4.2.91.2.1 auxiliary:** Restrict auxiliary exact dimension. Dimension\_tolerance to be a nominal value with no value\_limitation.

**4.2.91.2.2 theoretical:** Restrict theoretically exact dimension. Dimension\_tolerance to be a nominal value with no value\_limitation.

**4.2.91.2.3 user defined:** a description specified by the user.



### 4.2.91.3 dimension\_value

The dimension\_value specifies the total amount by which a specific dimension is permitted to vary.

### 4.2.91.4 limit

The limit specifies the tolerance value applied to the Dimension\_tolerance. The limit need not be specified for a particular Dimensional\_tolerance. See 4.3.134 for the application assertion.

### 4.2.91.5 placement

A placement defines how to place a Dimensional\_tolerance on the geometry of the part. See 4.3.135 for the application assertion.

### 4.2.91.6 significant\_digits

The significant\_digits specifies the number of decimal places indicating the accuracy of dimension or tolerance. Significant\_digits need not be specified for a particular Dimensional\_tolerance.

### 4.2.91.7 unit\_of\_measure

The unit\_of\_measure specifies the unit in which the quantity is expressed.

## 4.2.92 Direction\_element

A Direction\_element is a type of Shape\_element (see 4.2.285) that is a Shape\_aspect definition for a direction.

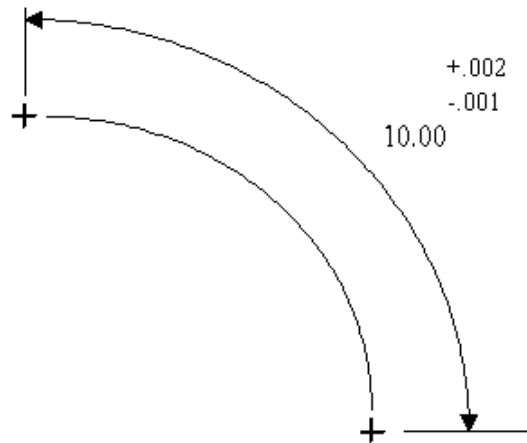
NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.71.

## 4.2.93 Distance\_along\_curve\_tolerance

A Distance\_along\_curve\_tolerance is a type of Location\_tolerance (see 4.2.169) that is the distance calculated between two elements along a path defined by a third element of geometry.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.72.

NOTE 2 Figure 11 illustrates the Distance\_along\_curve\_tolerance.



**Figure 11 — Distance\_along\_curve\_tolerance**

The data associated with a Distance\_along\_curve\_tolerance are the following:

- path;
- with\_curve\_direction.

#### **4.2.93.1 path**

The path specifies the shape that the tolerance applies to. See 4.3.136 for the application assertion.

#### **4.2.93.2 with\_curve\_direction**

The with\_curve\_direction specifies a Boolean value that indicates the direction along the element to apply the tolerance. A value of true indicates the tolerance value is applied from the start point of the curve to the end point of the curve. A value of false indicates that the direction does not matter.

## 4.2.94 Document\_assignment

A Document\_assignment is a mechanism to associate a document with an object, where the assigned document provides information about the object it is associated to. Each Document\_assignment may be a Customer\_design\_deficiency\_report (see 4.2.63), Customer\_tool\_design\_report (see 4.2.66), Part\_dimensioning\_standard (see 4.2.213), Product\_quality\_report (see 4.2.241), Proof\_report (see 4.2.252), Quality\_acceptance\_report (see 4.2.259), Simulation\_exception\_report (see 4.2.290), Simulation\_report (see 4.2.296) or a Simulation\_tooling\_report (see 4.2.301).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.29.

The data associated with a Document\_assignment are the following:

- assigned\_document;
- role.

### 4.2.94.1 assigned\_document

The assigned\_document specifies the Document\_file (see 4.2.95) that is used to provide information. See 4.3.137 for the application assertion.

### 4.2.94.2 role

The role specifies the meaning of the Document\_assignment.

## 4.2.95 Document\_file

A Document\_file is a file on a computer system or a paper document that make up a document representation. Each Document\_file is a Hardcopy (see 4.2.141) or a Digital\_file (see 4.2.88).

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.30.

The data associated with a Document\_file are the following:

- document\_properties;
- document\_type;
- file\_id;
- file\_location;
- version\_id.

### 4.2.95.1 document\_properties

The document\_properties specifies the characteristics of the Document\_file that relate to the format of the object. The document\_properties need not be specified for a particular Document\_file (see 4.2.95). See 4.3.138 for the application assertion.

### 4.2.95.2 document\_type

The document\_type specifies the word or the group of words that describe the kind of object the characteristics are provided for. The document\_type need not be specified for a particular Document\_file.

The values of the document\_type may be one of the following:

- check plan;
- design\_reference\_data;
- geometry;
- NC data;
- process plan\_data;
- sample data.

NOTE See 4.2.95.2.1 to 4.2.95.2.6 for the definition of each allowable value for document\_type.

**4.2.95.2.1 check plan:** the document represents quality control planning data.

**4.2.95.2.2 design\_reference\_data:** the document represents a technical drawing.

**4.2.95.2.3 geometry:** the document represents a shape model.

**4.2.95.2.4 NC data:** the document represents numerical control data.

**4.2.95.2.5 process\_plan\_data:** the document represents process planning data.

**4.2.95.2.6 sample data:** the document represents measured data.

### 4.2.95.3 file\_id

The file\_id specifies the identifier which is used to locate the file either on a computer system or in a repository of paper documents.

#### 4.2.95.4 file\_location

The file\_location specifies the information necessary to locate the file in an external storage area. A file\_location need not be specified for a particular Document\_file. See 4.3.139 for the application assertion.

#### 4.2.95.5 version\_id

The version\_id specifies the identification of the version that distinguishes one Document\_file object from other versions of Document\_file objects with the same file\_id. The version\_id need not be specified for a particular Document\_file.

### 4.2.96 Document\_file\_properties

The Document\_file\_property specifies characteristics of a Document\_file (see 4.2.95) that specify the format of the object. At least one of the optional attributes shall be specified for each instance of this object.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.31.

The data associated with a Document\_file\_properties are the following:

- content\_country\_code;
- content\_detail\_level;
- content\_geometry\_type;
- content\_language\_code;
- creating\_interface;
- creating\_operating\_system;
- creating\_system;
- data\_format;
- file\_size;
- format\_character\_code;
- page\_count.

### **4.2.96.1 content\_country\_code**

The `content_country_code` specifies the country, as addition to the language, according to the alpha-2 code specified in ISO 3166-1. The `country_code` need not be specified for a particular `content_language_code`.

EXAMPLE Possible values for `country_code` may be, 'GB' for the United Kingdom or 'US' for the United States of America.

### **4.2.96.2 content\_detail\_level**

The `content_detail_level` specifies the level of detail that the `Document_file` (see 4.2.95) provides. The `content_detail_level` need not be specified for a particular `Document_content_property`.

The values of the `content_detail_level` may be one of the following:

- rough 3d shape;
- rounded edges.

NOTE See 4.2.96.2.1 to 4.2.96.2.2 for the definition of each allowable value for `content_detail_level`.

**4.2.96.2.1** rough 3d shape: the 3D shape model without edge rounds and fillets.

**4.2.96.2.2** rounded edges: the 3D shape model with edge rounds and fillets.

### **4.2.96.3 content\_geometry\_type**

The `content_geometry_type` specifies the kind or kinds of geometry that an object contains. The `content_geometry_type` need not be specified for a particular `Document_content_property`.

The values of the `content_geometry_type` may be one of the following:

- assembly;
- assembly with mating elements;
- closed volume;
- drawing derived from 3D data;
- drawing related to 3D data;
- solid and surface model;
- solid model;
- surface model;

- 2D drawing;
- 2D shape;
- 3D wireframe model.

NOTE See 4.2.96.3.1 to 4.2.96.3.11 for the definition of each allowable value for `content_geometry_type`.

**4.2.96.3.1** 2D assembly: the document contains an assembly structure with reference to the assembled components and their transformation matrices.

**4.2.96.3.2** assembly with mating elements: the document contains an assembly structure including the mating components only, such as screws or rivets, with exact positioning information. This assembly representation is intended to be overlaid with the assembly structure for the main components.

**4.2.96.3.3** 2D closed volume: The document contains a 3D shape model in closed body topological surface representation.

**4.2.96.3.4** drawing derived from 3D data: the document contains a technical drawing that has been derived from a 3D shape model.

**4.2.96.3.5** drawing related to 3D data: the document contains a technical drawing that visualizes a 3D shape model and possibly establishes associative links to the 3D shape model.

**4.2.96.3.6** solid model: the document contains a 3D shape model in advanced boundary representation.

**4.2.96.3.7** solid and surface model: the document contains a 3D shape model in surface and advanced boundary representation.

**4.2.96.3.8** 2D surface model: the document contains a 3D shape model in surface representation.

**4.2.96.3.9** 2D shape: the document contains a 2D shape model or contours only.

**4.2.96.3.10** 2D drawing: the document contains a technical drawing without 3D shape representation.

**4.2.96.3.11** 3D wireframe model: the document contains a 3D shape model in wireframe representation.

#### **4.2.96.4 content\_language\_code**

The `content_language_code` specifies the language of the text information in the Alpha-3 bibliographic code specified in ISO 639-2. The `content_language_code` need not be specified for a particular `Document_file`.

EXAMPLE Possible values for `language_code` may be, 'eng' for English, 'fre' for French, 'rus' for Russian, or 'ger' for German.

#### **4.2.96.5 creating\_interface**

The `creating_interface` specifies the computer application used to create the `Document_file` (see 4.2.95) object. The `creating_interface` need not be specified for a particular `Document_creation_property`.

EXAMPLE 1 'Postscript Printer Driver' is an example for a creating interface.

EXAMPLE 2 'SYSTEM-C-STL' is an example for a creating interface in the case of a stereo lithographic model.

#### **4.2.96.6 creating\_operating\_system**

The `creating_operating_system` specifies the operating system that is used to execute the computer application that created the characterized object. The `creating_operating_system` need not be specified for a particular `Document_file_property`.

#### **4.2.96.7 creating\_system**

The `creating_system` specifies the computer application or the machine which is used to create the object that is characterized. The `creating_system` need not be specified for a particular `Document_file_property`.

#### **4.2.96.8 data\_format**

The `data_format` specifies the convention that was used to structure the information in the characterized object. A `data_format` need not be specified for a particular `Document_file`. The `data_format` need not be specified for a particular `Document_format_property`.

The values of the `data_format` may be one of the following:

- DXF;
- IGES;
- ISO 10303-203;
- ISO 10303-214;
- ISO 10303-224;
- ISO 10303-238;
- ISO 6983;
- ISO 13399;



- TIFF CCITT GR4;
- VDAFS;
- VOXEL.

NOTE See 4.2.96.8.1 to 4.2.96.8.10 for the definition of each allowable value for data\_format.

**4.2.96.8.1** DXF: the document contains data in Drawing Exchange File format.

**4.2.96.8.2** IGES: the document contains data in Initial Graphics Exchange Specification format.

**4.2.96.8.3** ISO 10303-203: the document contains data in ISO 10303-203 format.

**4.2.96.8.4** ISO 10303-214: the document contains data in ISO 10303-214 format.

**4.2.96.8.5** ISO 10303-224: the document contains data in ISO 10303-214 format.

**4.2.96.8.6** ISO 10303-238: the document contains data in ISO 10303-238 format.

**4.2.96.8.7** ISO 6983 : M and G codes for NC controllers.

**4.2.96.8.8** TIFF CCITT GR4: the document contains data in TIFF CCITT GR4 format.

**4.2.96.8.9** VDAFS: the document contains data in VDAFS format.

**4.2.96.8.10** VOXEL: the document contains data in VOXEL format.

### **4.2.96.9 file\_size**

The file\_size specifies the size of a digitally stored document. The file\_size shall only be applied in cases where the Document\_size\_property is referred by a Document\_file (see 4.2.95). The file\_size need not be specified for a particular Document\_size\_property.

EXAMPLE '15021 Bytes' and 'less than 500 Bytes' are examples for a file\_size.

### **4.2.96.10 format\_character\_code**

The format\_character\_code specifies the character code that is used in the characterized object. A format\_character\_code need not be specified for a particular Document\_file.

The values of the format\_character\_code may be one of the following:

- binary;
  - IEC 61286;
  - ISO/IEC 646;
- © ISO 2008 — All rights reserved

ISO 10303-223:2008 (E)

- ISO 3098-2;
- ISO/IEC 6937;
- ISO 8859-1;
- ISO/IEC 10646.

NOTE See 4.2.96.10.1 to 4.2.96.10.7 for the definition of each allowable value for `format_character_code`.

**4.2.96.10.1** `binary`: the document contains data in binary format.

**4.2.96.10.2** `IEC 61286`: the coded character set used to encode the document data according to IEC 61286.

**4.2.96.10.3** `ISO/IEC 646`: the coded character set used to encode the document data according to ISO/IEC 646.

NOTE The character set in ISO/IEC 646 is identical to the character set commonly known as ASCII

**4.2.96.10.4** `ISO 3098-2`: the coded character set used to encode the document data is according to ISO 3098-2.

**4.2.96.10.5** `ISO/IEC 6937`: the coded character set used to encode the document data is according to ISO/IEC 6937.

**4.2.96.10.6** `ISO 8859-1`: the coded character set used to encode the document data according to ISO 8859-1.

NOTE The character set in ISO 8859-1 is identical to the character set commonly known as LATIN-1.

**4.2.96.10.7** `ISO/IEC 10646`: the coded character set used to encode the document data according to ISO/IEC 10646.

### **4.2.96.11** `page_count`

The `page_count` specifies the number of pages of the application object the `Document_size_property` is referred by. The `page_count` shall only be used in cases where the `Document_size_property` is referred by a `Hardcopy` (see 4.2.141). The `page_count` need not be specified for a particular `Document_size_property`.

EXAMPLE '42 pages' and 'more than 1 page' are examples of a `page_count`.

## 4.2.97 Document\_file\_relationship

A Document\_file\_relationship is a relationship between two Document\_file (see 4.2.95) objects. It specifies that the related Document\_file is referenced from the relating Document\_file.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.32.

EXAMPLE A service manual may contain graphics for explanatory reasons. In this case the Document\_file objects that contain the graphics are referenced as related from the Document\_file object that contains the body of the service manual with relation\_type 'reference'.

The data associated with a Document\_file\_relationship are the following:

- previous\_file;
- succeeding\_file.

### 4.2.97.1 previous\_file

The previous\_file specifies the Document\_file with the highest precedence. See 4.3.140 for the application assertion.

### 4.2.97.2 succeeding\_file

The succeeding\_file specifies the Document\_file with a lesser precedence. See 4.3.140 for the application assertion.

## 4.2.98 Drafted\_surface

A Drafted\_surface is a type of Shape\_element (see 4.2.285) that specifies the surface(s) that has a slant defined from the parting line to permit an object to be easily removed from a mould. The data associated with a Drafted\_surface are the following:

NOTE the slanted surfaces will always increase the dimensional size at the parting line.

- draft;
- shape\_element.

### **4.2.98.1 draft**

The draft defines the slant or taper tending away from the parting line given to those Drafted\_surfaces of a pattern or core box that would lie in the direction in which the pattern, core box, or its component parts are drawn from the sand. See 4.3.141 for the application assertion.

NOTE 1 the draft is most often defined as an angle in degrees or sometimes as inches per foot.

NOTE 2 This application object is harmonized with the draft definition from glossary of metalcasting terms from AFS [2].

EXAMPLE An example of draft may be 3 degrees or 1.25 in. per 1 ft.

### **4.2.98.2 shape\_element**

The shape\_element specifies the particular portion of the part shape to which draft shall be applied. See 4.3.142 and 4.3.143 for the application assertion.

## **4.2.99 Dryer**

A Dryer is a type of Core\_equipment (see 4.2.54) that is used to bake oil sand cores in an oven before they are place cores in the mould. The data associated with a Dryer are the following:

— material\_type.

### **4.2.99.1 material\_type**

The material\_type specifies the word or group of words that describes the substance from which the Dryer is made.

### **4.2.100 Edge\_shape\_element**

A Edge\_shape\_element is a type of Shape\_element (see 4.2.285) that is a Shape\_aspect definition for a Edge.

### **4.2.101 Ejector\_box**

The Ejector\_box is a type of Ejector\_design\_feature (see 4.2.102) that the frame that houses a function ejector system and is secured to the moving half core box or die tool. The data associated with an Ejector\_box are the following:

— box\_size;

— oriented\_to\_movable\_die;

— plate\_definition.

### 4.2.101.1 box\_size

The `box_size` specifies the dimensional values of the container size used to retain the ejector system. See 4.3.145 for the application assertion.

### 4.2.101.2 oriented\_to\_movable\_die

The `oriented_to_movable_die` specifies the specific mould type for which an `Ejector_box` is attached. See 4.3.147, 4.3.148, 4.3.149 for the application assertion.

### 4.2.101.3 plate\_definition

The `plate_definition` specifies one or more `Ejector_plate`(s) assembled for use inside the `Ejector_box`. See 4.3.144 for the application assertion.

## 4.2.102 Ejector\_design\_feature

The `Ejector_design_feature` is a type of `Feature` (see 4.2.124) that is required in the design of an ejector system. An `Ejector_design_feature` is either a `Ejector_box` (see 4.2.101), `Ejector_plate` (see 4.2.105), `Stop_pin` (see 4.2.313), `Reset_or_return_pins` (see 4.2.266), `Ejector_pins` (see 4.2.104) or `Pin_tips` (see 4.2.230). The data associated with a `Ejector_design_feature` are the following:

- `placement`.

### 4.2.102.1 placement

The `placement` specifies the position and orientation of a `Ejector_design_feature` relative to the base shape for a part. See 4.3.150 for the application assertion.

## 4.2.103 Ejector\_equipment

A `Ejector_equipment` is a type of `Equipment` (see 4.2.110) that is used in producing the ejector system.

The data associated with a `Ejector_equipment` are the following:

- `ejector_characteristic`.

### 4.2.103.1 ejector\_characteristic

The `Ejector_system` specifies the pin and plate information required for the ejectors. See 4.3.151 for the application assertion.

NOTE Ejector pins may be necessary to release a core from a core box when there are deep draws in the core box or minimum draft permitted.

### **4.2.104 Ejector\_pins**

The Ejector\_pins is a type of Ejector\_design\_feature (see 4.2.102) that specifies each pin used in the Ejector\_system for size, shape, and location.

NOTE This application object is harmonized with the ejector pins definition from glossary of metalcasting terms from AFS [2].

The data associated with a Ejector\_pins are the following:

- pin\_dimensions;
- thread;
- tip\_definition.

#### **4.2.104.1 pin\_dimensions**

The pin\_dimensions specifies the length, diameter, and taper if required for the pin. See 4.3.153 for the application assertion.

#### **4.2.104.2 tip\_definition**

The tip\_definition specifies the shape for the tip of the ejector pin at the point of intersection with the core box mould cavity shape area shape area. See 4.3.154 for the application assertion.

#### **4.2.104.3 thread**

The thread specifies the type and size of the pin thread. See 4.3.152 for the application assertion.

NOTE the type and size will be one of the standard catalogue threads.

### **4.2.105 Ejector\_plate**

The Ejector\_plate is a type of Ejector\_design\_feature (see 4.2.102) that specifies the plate size, shape, and the pin locations required for the ejector system assembly.

The data associated with a Ejector\_plate are the following:

- ejector\_pin\_definition;
- plate\_height;
- plate\_material;

- plate\_shape;
- reset\_push\_pin\_definition;
- stop\_pin\_definition.

#### **4.2.105.1 ejector\_pin\_definition**

The `ejector_pin_definition` specifies the requirements for each pin needed in the ejector system assembly. See 4.3.155 for the application assertion.

#### **4.2.105.2 plate\_height**

The `plate_height` specifies the thickness of each plate to be used in the ejector system. See 4.3.157 for the application assertion.

#### **4.2.105.3 plate\_material**

The `plate_material` specifies the substance used for the ejector plate.

NOTE See 4.2.105.3.1 for the definition of each allowable value for `plate_material_type`.

**4.2.105.3.1 aluminum:** an available commercial grade aluminum plate that is straight and warp free.

#### **4.2.105.4 plate\_shape**

The `plate_shape` specifies the implicit or explicit shape of the ejector plate. The implicit shape is the size of the rectangle or circle together with the height or thickness of a plate to be used. The explicit shape is the geometric shape of a plate to be used. See 4.3.156 for the application assertion.

#### **4.2.105.5 reset\_push\_pin\_definition**

The `reset_push_pin_definition` specifies the size, location, and when needed the spring information of the pins to be used. See 4.3.158 for the application assertion.

NOTE for reset pins a spring description will be needed.

#### **4.2.105.6 stop\_pin\_definition**

The `stop_pin_definition` specifies the requirement information for each pin needed to stop an `Ejector_plate` in the ejector system assembly. The `stop_pin_definition` need not be specified for a particular `Ejector_plate`. See 4.3.159 for the application assertion.

## 4.2.106 Ejector\_system

The Ejector\_system specifies the pin and plate information required for the ejectors.

NOTE 1 Ejector pins may be necessary to release a core from a core box when there are deep draws in the core box or minimum draft permitted.

NOTE 2 This application object is harmonized with the ejector object from ISO 8062-1:2007, Clause 2.8.

The data associated with a Ejector\_system are the following:

- plate\_definition;
- system\_operated\_by.

### 4.2.106.1 plate\_definition

The plate\_definition specifies the material needed to anchor the pins for the ejector system. See 4.3.160 for the application assertion.

### 4.2.106.2 system\_operated\_by

The system\_operated\_by specifies how the ejector system is activated.

NOTE See 4.2.106.2.1 to 4.2.106.2.4 for the definition of each allowable value for system\_operated\_by.

**4.2.106.2.1 hydraulics:** a mechanism operated by liquid under pressure to cause an object to move some specified distance.

**4.2.106.2.2 pneumatic:** a mechanism operated by compressed air to cause an object to move some specified distance.

**4.2.106.2.3 spring\_loaded:** a mechanism operated by compressing one or more helical compression springs placed under pressure to cause an object to move some specified distance when released.

**4.2.106.2.4 simple\_mechanical:** a mechanism operated by one set of pins pushing in one direction while another set of pins push in the opposite direction to cause an object to move some specified distance.

## 4.2.107 Elevated\_temperature\_service

An Elevated\_temperature\_service are the designed temperature requirements that the cast part will be subjected to for in service applications. The data associated with a Elevated\_temperature\_service are the following:

- maximum\_cyclic\_temperature;
- maximum\_temperature;



- `minimum_cyclic_temperature`;
- `service_descriptions`.

#### **4.2.107.1 maximum\_cyclic\_temperature**

The `maximum_cyclic_temperature` is the highest designed operating temperature with units for periodic application for which the cast part is designed. See 4.3.162 for the application assertion.

#### **4.2.107.2 maximum\_temperature**

The `maximum_temperature` is the highest designed operating temperature with units for the in-service application for the cast part. See 4.3.162 for the application assertion.

#### **4.2.107.3 minimum\_cyclic\_temperature**

The `minimum_cyclic_temperature` is the lowest designed operating temperature with units for periodic application for which the cast part is designed. See 4.3.162 for the application assertion.

#### **4.2.107.4 service\_descriptions**

A `service_descriptions` describes the conditions where elevated temperatures problematic to the cast part as designed. The `service_descriptions` need not be specified for a particular `Elevated_temperature_service` for the cast part. See 4.3.161 for the application assertion.

### **4.2.108 Engineering\_change\_order**

An `Engineering_change_order` is an authorization for modification of the product data that will result in a new process plan for a part.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.74.

NOTE 2 These `Engineering_change_orders` apply only to changes that effect cast parts.

The data associated with an `Engineering_change_order` are the following:

- `change_order_number`;
- `new_version`.

#### **4.2.108.1 change\_order\_number**

The `change_order_number` specifies a unique identification of the `Engineering_change_order`.

#### **4.2.108.2 new\_version**

The `new_version` specifies the current version of a Part which has an effected change made by the `Engineering_change_order`. There may be more than one `new_version` of an `Engineering_change_order`. See 4.3.163 for the application assertion.

#### **4.2.109 Engineering\_change\_proposal**

An `Engineering_change_proposal` is a document that describes potential modifications to a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.75.

The data associated with an `Engineering_change_proposal` are the following:

- `change_proposal_number`;
- `incorporated_proposal`.

##### **4.2.109.1 change\_proposal\_number**

The `change_proposal_number` specifies a unique identification of the `Engineering_change_proposal`.

##### **4.2.109.2 incorporated\_proposal**

The `incorporated_proposal` specifies the change proposals that describe the modifications to the Part. There may be more than one `incorporated_proposal` for an `Engineering_change_proposal`. See 4.3.164 for the application assertion.

#### **4.2.110 Equipment**

An `Equipment` is the machinery necessary to execute a casting activity. Each `Equipment` is either a `Core_equipment` (see 4.2.54), `Ejector_equipment` (see 4.2.103), `Melting_equipment` (see 4.2.194), `Moulding_equipment` (see 4.2.201), `Shakeout` (see 4.2.281), or `Finishing` (see 4.2.127).

NOTE This application object is harmonized with the machine object from ISO 10303-240:2005, Clause 4.2.68.

The data associated with an `Equipment` are the following:

- `controlled_by`;
- `equipment_model`;
- `equipment_name`;

- identification;
- location.

#### **4.2.110.1 controlled\_by**

The `controlled_by` specifies the Controller (see 4.2.46) associated with this specific Equipment. A `controlled_by` need not be specified for a particular Equipment. See 4.3.165 for the application assertion.

#### **4.2.110.2 equipment\_model**

The `equipment_model` specifies the unique identification, the design version identification assigned by the machine builder, for a set of machines that all have the same characteristics.

NOTE 1419An is an example of equipment model for a 14 x19 Hunter automatic moulding machine

#### **4.2.110.3 equipment\_name**

The equipment name specifies the label used to identify the equipment.

NOTE The equipment name may be the make and model of equipment.

#### **4.2.110.4 identification**

The identification specifies the label or serial number used to uniquely identify used to identify a specific piece of machinery. The identification need not be specified for a particular Equipment.

#### **4.2.110.5 location**

The location specifies where in the factory, a particular piece of equipment is located. The location need not be specified for a particular Equipment. See 4.3.167 for the application assertion.

### **4.2.111 Equipment\_process**

A `Equipment_process` is a type of `Manufacturing_process` (see 4.2.180) that specifies the type of machine, the type of manufacturing setup, and a sequential list of `Casting_activity` (see 4.2.21) required to perform an automated machining process on a machine. The data associated with a `Equipment_process` are the following:

- `required_equipment`.

#### **4.2.111.1 required\_equipment**

The `required_machine` specifies the Equipment (see 4.2.110) to be used for a specific machine setup. See 4.3.167 for the application assertion.

## **4.2.112 Equipment\_setting\_record**

A `Equipment_setting_record` is a type of `Activity_result_record` (see 4.2.3) that is a record of a value for an equipment operating available during a particular execution of a `Activity` (see 4.2.1).

NOTE A `Equipment_setting_record` may be used for later analysis to determine the effect of an equipment operating variable on the quality of the part.

The data associated with a `Equipment_setting_record` are the following:

- `actual_setting`;
- `recorded_usage`.

### **4.2.112.1 actual\_setting**

The `actual_setting` specifies the measured value for a `Equipment_usage` (see 4.2.115). See 4.3.169 for the application assertion.

### **4.2.112.2 recorded\_usage**

The `record_usage` specifies the `Property_parameter` (see 4.2.256) that the `Equipment_setting_record` is being applied. See 4.3.168 for the application assertion.

## **4.2.113 Equipment\_setup**

A `Equipment_setup` is a type of `Activity_setup` (see 4.2.1) that is the activity of preparing equipment conditions on a specific piece of equipment prior to performing casting operations.

NOTE This application object is harmonized with the `machine_setup` object from ISO 10303-240:2005, Clause 4.2.70.

The data associated with a `Equipment_setup` are the following:

- `identifies`;
- `setup_instruction`.

### **4.2.113.1 identifies**

The `identifies` specifies the `Equipment` (see 4.2.110) that is being setup for an `Activity` (see 4.2.1). See 4.3.170 for the application assertion.

### **4.2.113.2 setup\_instruction**

The `setup_instruction` specifies the word, or group of words, that explain the work performed to setup the `Equipment`.

### 4.2.114 Equipment\_size

The Equipment\_size specifies a dimensional value with the appropriate units assigned to a particular item or detail. The data associated with an Equipment\_size are the following:

- item\_height
- item\_width;
- item\_length;
- item\_draft;
- item\_diameter.

#### 4.2.114.1 item\_height

The item\_height is the measured perpendicular distance with appropriate units from the base of an item to its top. See 4.3.246 for the application assertion.

NOTE sometimes height and thickness are used interchangeably for an item.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 2 inch value is the item\_height.

#### 4.2.114.2 item\_width

The item\_width is the measured perpendicular distance with appropriate units from one side of an item to its side opposite. See 4.3.246 for the application assertion.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 8 inch value is the item\_width.

#### 4.2.114.3 item\_length

The item\_length is the measured perpendicular distance with appropriate units from one end of an item to its end opposite. See 4.3.246 for the application assertion.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 3 feet value is the item\_length.

#### 4.2.114.4 item\_draft

The item\_draft is the measured value with units of the slant or taper required to remove an item from the mould. See 4.3.246 for the application assertion.

NOTE the slant or taper is most often defined as a measured angle with units in degrees.

#### **4.2.114.5 item\_diameter**

The `item_diameter` is the measured distance with appropriate units defined as 2 times the radius value that has the same units. See 4.3.246 for the application assertion.

NOTE The `item_diameter` may also be defined as the measured distance with appropriate units of a straight line that has two points on the circumference of a circle or sphere such that the straight line passes through the centre of the circle or sphere.

EXAMPLE A round bar of steel that measures 2 inches by 3 feet in size. The 2 inch value is the `item_diameter`.

#### **4.2.115 Equipment\_usage**

The `Equipment_usage` is the context or circumstance in which an equipment is to be used and the parameter values for its use in that context or circumstance. The data associated with an `Equipment_usage` are the following:

- `equipment_usage_context`;
- `equipment_parameter`.

##### **4.2.115.1 equipment\_usage\_context**

An `equipment_usage_context` specifies the conditions or circumstances for which the equipment is to be used. See 4.3.173 for the application assertion.

##### **4.2.115.2 equipment\_parameter**

The `equipment_parameter` specifies the operational values required to operate the equipment. See 4.3.172 for the application assertion.

#### **4.2.116 Executable**

The `Executable` is a type of `External_schema_definition` (see 4.2.118) that initiate actions on a machine and shall be arranged in a defined order.

NOTE Executable definitions are derived from ISO 10303:238:2007.

The data associated with an `Executable` the following:

- `executable_id`.

### 4.2.116.1 executable\_id

The executable\_id specifies the descriptive name of the ISO 10303 part entity being referenced.

EXAMPLE 1 An example of an executable\_id would be #10, where #10 in the reference\_schema may be:#10=WORKPLAN('MAIN WORKPLAN',(#11,#12,#13,#14,#15),\$,#36).

EXAMPLE 2 An example of an executable\_id would be #10, where #10 in the reference\_schema may be:#10=PROGRAM\_STOP.

### 4.2.117 External\_file\_identification

An External\_file\_identification specifies the location of a file in an external storage system.

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.37.

The data associated with an External\_file\_id\_and\_location are the following:

- external\_file\_id;
- external\_file\_location.

#### 4.2.117.1 external\_file\_id

The external\_id specifies the identifier of a document in an external storage system. An external\_id need not be specified for a particular External\_file\_identification

EXAMPLE An example of an external\_file\_id may be 'specification.txt'.

#### 4.2.117.2 external\_file\_location

The location specifies the location of the Document\_file in the external storage system.

EXAMPLE 'D:n project1n ' and '/projects/project1/'.

The combination of file id and its location taken from the previous examples are 'D:n project1n specification.txt' and '/projects/project1/specification.txt' respectively

### 4.2.118 External\_schema\_definition

An External\_schema\_definition is a reference mechanism to specify external documents that are associated with objects defined in this part of ISO 10303. These documents are other ISO 10303 parts that may contain data that are not in scope of this part of ISO 10303.

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.38.

ISO 10303-223:2008 (E)

The data associated with an `External_schema_definition` are the following:

- `reference_schema`.

#### **4.2.118.1 reference\_schema**

The `reference_schema` specifies a file being referenced that conforms to the ISO 10303 schema in digital format. See 4.3.174 for the application assertion.

EXAMPLE An example of a `reference_schema` is a file that conforms to the ISO 10303-238 schema.

#### **4.2.119 Externally\_defined\_representation**

An `Externally_defined_representation` is used to identify a piece of product data whose definition is provided within an external specification or document.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.39.

The data associated with an `Externally_defined_representation` are the following:

- `identified_by`;
- `location`;
- `placement`.

##### **4.2.119.1 identified\_by**

The `identified_by` specifies the part being referenced, that is contained in an external library. See 4.3.175 for the application assertion.

##### **4.2.119.2 location**

The `location` specifies the orientation of the external product data within the geometric domain of the product data of this part of ISO 10303. A `location` need not be specified for a particular `Externally_defined_representation`.

##### **4.2.119.3 placement**

The `placement` specifies the positioning of the external product data within the geometric domain of the product data of this part of ISO 10303. A `placement` need not be specified for a particular `Externally_defined_representation`.



### 4.2.120 Externally\_defined\_size\_dimension

An Externally\_defined\_size\_dimension is a type of Size\_tolerance (see 4.2.304) that is used to identify a size dimension whose definition is provided within an external specification or document.

NOTE This application object is harmonized with the same from ISO 10303-224:2006, Clause 4.277.

The data associated with an Externally\_defined\_size\_dimension are the following:

- path;
- tolerance\_class;
- tolerance\_definition.

#### 4.2.120.1 path

The path specifies the shape that the tolerance applies to. A Externally\_defined\_size\_dimension may but need not require a path. See 4.3.176 for the application assertion.

#### 4.2.120.2 tolerance\_class

The tolerance\_class specifies a type of size tolerance that is being defined by in external document.

#### 4.2.120.3 tolerance\_definition

The tolerance\_definition specifies a file being referenced that contains information about the type of tolerance. See 4.3.177 for the application assertion.

### 4.2.121 Fabrication

In Fabrication is a type of Non\_casting\_operations (see 4.2.202) that identifies the activity of joining, usually by welding, of two or more parts to produce a finished assembly. The components of the assembly may be a combination of cast and wrought materials.

### 4.2.122 Face\_shape\_element

A Face\_shape\_element is a type of Shape\_element (see 4.2.285) that is a Shape\_aspect definition for a Face.

NOTE This application object is harmonized with the same from ISO 10303-224:2006, Clause 4.2.79.

### **4.2.123 Face\_shape\_element\_relationship**

A `Face_shape_element_relationship` defines the sequence in which `face_shape_element` objects are applied. The `Face_shape_element_relationship` defines which face is the preceding face and which face is the succeeding face.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.80.

The data associated with a `Face_shape_element_relationship` are the following:

- predecessor;
- successor.

#### **4.2.123.1 predecessor**

The predecessor specifies the `Face_shape_element` (see 4.2.122) with the highest precedence. See 4.3.178 for the application assertion.

#### **4.2.123.2 successor**

The successor specifies the `Face_shape_element` (see 4.2.122) with the lesser precedence. See 4.3.178 for the application assertion.

### **4.2.124 Feature**

A `Feature` is a type of `Shape_element` (see 4.2.285) that defines the characteristic shape for a specific design feature required to produce a casting relative to the `Design_part` (see 4.2.76). Each `Feature` is either a `Manufacturing_design_feature` (see 4.2.176), `Casting_design_feature` (see 4.2.22), `Ejector_design_feature` (see 4.2.102), `Sand_casting_design_feature` (see 4.2.276), `Gating_design_feature` (see 4.2.133), `Die_design_feature` (see 4.2.83) or `Investment_design_feature` (see 4.2.161).

NOTE This application object is harmonized with the feature of size object from ISO 8062-1:2007, Clause 4.1.

### **4.2.125 Feeder**

The `Feeder` is a type of `Gating_design_feature` (see 4.2.133) that provide a channel for the molten metal to flow to the riser and from the riser to the casting. The channel shape is formed by a pattern configured to the gating system requirements.

NOTE This application object is harmonized with the feeder definition from glossary of metalcasting terms from AFS [2].

The data associated with a Feeders are the following:

- parametric\_shape.
- runner\_connection.
- transition.

#### **4.2.125.1 parametric\_shape**

The parametric\_shape specifies the implicit or explicit shape of the ejector plate. The implicit shape is the size of the rectangle or circle together with the height or thickness of a feeder to be used. The explicit shape is the geometric shape of a feeder to be used. See 4.3.181 for the application assertion.

#### **4.2.125.2 runner\_connection**

The runner\_connection specifies the contact information for each connection to a riser from the runner. See 4.3.180 for the application assertion.

#### **4.2.125.3 transition**

The transition specifies the radius that transitions the side of the channel to the top of the channel. See 4.3.179 for the application assertion.

### **4.2.126 Filter**

A Filter is a type of Gating\_design\_feature (see 4.2.133) that prevents the entrance of slag and other extraneous material from entering the mould cavity.

NOTE 1 A filter is a small perforated core like strainer located in the gating system to filter the molten metal.

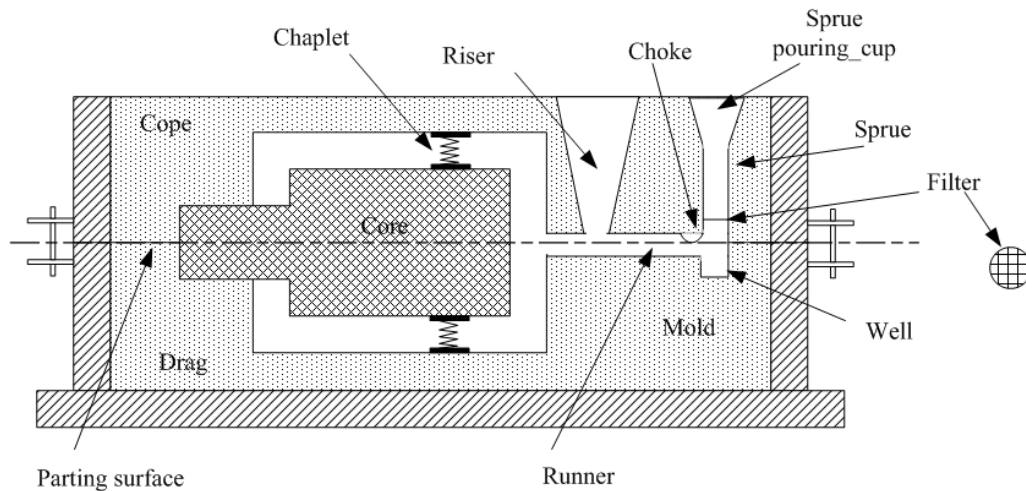
NOTE 2 Figure 12 illustrates a Gating\_system with a Filter.

The data associated with a Filter are the following:

- filter\_size.
- type\_of\_filter.

#### **4.2.126.1 type\_of\_filter**

The type\_of\_filter specifies the style of filtration or the material of the filter to be used.



**Figure 12 — Filter connection in a Gating\_system**

#### 4.2.126.2 filter\_size

The filter\_size specifies the dimensions of the filter. See 4.3.182 for the application assertion.

#### 4.2.127 Finishing

The Finishing is a type of Equipment (see 4.2.110) that provides value added operations to a cast part as requested by the customer.

EXAMPLE Equipment for painting, coating, welding, machining, and others are equipment for value added operations.

#### 4.2.128 Finishing\_or\_machining\_operations

The Finishing\_or\_machining\_operations are value added operations to the casting(s) before delivery of the cast part(s) to the customer. The data associated with a Finishing\_or\_machining\_operations are the following:

- comments;
- finishing\_operations;
- machining\_operation.

### 4.2.128.1 comments

The comments specifies specific parameter data for the required service the cast part is designed to withstand. The comments need not be specified for a particular Casting\_service\_data for the cast part. See 4.3.183 for the application assertion.

### 4.2.128.2 finishing\_operations

The finishing\_operation specifies one or more finishing activities or procedures on the cast part as per the design. The finishing\_operation need not be specified for a particular Finishing\_and\_machining\_operation for the cast part. See 4.3.184 for the application assertion.

### 4.2.128.3 machining\_operation

The machining\_operation specifies one or more machining activities or procedures on the cast part as per the design. The machining\_operation need not be specified for a particular Finishing\_and\_machining\_operation for the cast part. See 4.3.184 for the application assertion.

### 4.2.129 First\_article

A First\_article is a type of Casting\_verification (see 4.2.28 that is the first group of cast parts that are produced from tooling for the first time and the tooling has not been certified for production usage. The data associated with a First\_article are the following:

- customer\_approval;
- customer\_approved;
- delivery\_date;
- number\_of\_pieces;

#### 4.2.129.1 customer\_approval

A customer\_approval identifies the individual in the customer organization or their agent by person and organization who approved the inspected castings or the tooling that produced the castings. See 4.3.185 for the application assertion.

#### 4.2.129.2 customer\_approved

The customer\_approved specifies a sample part as the Master\_sample for which all production parts of that Design\_part will be evaluated. The customer\_approved need not be specified for a particular First\_article. See 4.3.187 for the application assertion.

### **4.2.129.3 delivery\_date**

The `delivery_date` specifies the day, month, and year the first article casting are to be delivered to the customer. See 4.3.186 for the application assertion.

### **4.2.129.4 number\_of\_pieces**

The `number_of_pieces` specifies the total number of cast parts included in the first article inspection package.

### **4.2.130 Fixture\_tab**

A `Fixture_tab` is a type of `Casting_design_feature` (see 4.2.22) that is an appendage added to the casting to secure or locate the cast part for machining.

NOTE A fixture tab may be added by request from machining and the tab remain as part of the original or be removed during machining operations.

### **4.2.131 Flask**

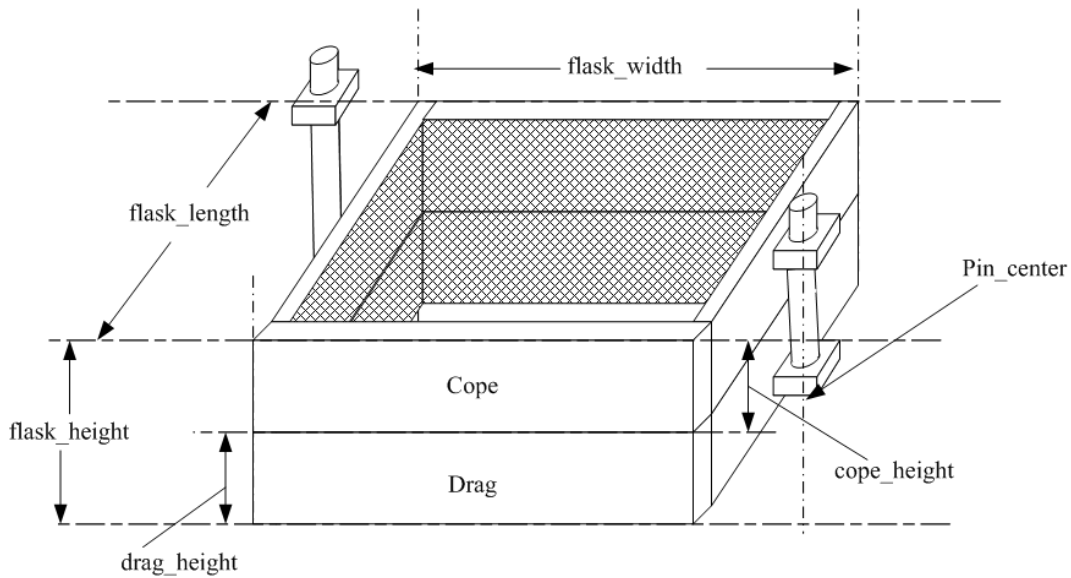
A `Flask` is a type of `Sand_casting_design_feature` (see 4.2.276) that is a frame, typically wooden or metal, that consists of two or more sections used to enclose the material in which the mould is formed.

NOTE 1 Figure 13 illustrates a Flask.

NOTE 2 This application object is harmonized with the flask definition from glossary of metalcasting terms from AFS [2].

The data associated with a Flask are the following:

- `cope_height`;
- `drag_height`;
- `flask_length`;
- `flask_width`;
- `generates`;
- `pin_center_definition`;
- `type_of_flask`.



**Figure 13 — Flask**

#### 4.2.131.1 cope\_height

The **cope\_height** specifies the distance from the parting line to the top of the mould. See 4.3.188 for the application assertion.

NOTE The cope height is the required distance needed to fill the mould cavity with molten metal.

#### 4.2.131.2 drag\_height

The **drag\_height** specifies the distance from the parting line to the bottom of the mould. See 4.3.188 for the application assertion.

NOTE The drag depth is the required distance needed to stabilize the mould cavity with molten metal.

#### 4.2.131.3 flask\_length

The **flask\_length** specifies the inside of the flask frame, measured at the parting line. See 4.3.188 for the application assertion.

NOTE The flask length is the required distance needed to stabilize the mould cavity with molten metal.

#### **4.2.131.4 flask\_width**

The `flask_width` specifies the inside of the flask frame, measured at the parting line. See 4.3.188 for the application assertion.

NOTE The flask width is the required distance needed to stabilize the mould cavity with molten metal.

#### **4.2.131.5 generates**

The `generates` specifies a mould that is produced from the Flask; after filled with sand and the pattern has been removed. See 4.3.191 for the application assertion.

#### **4.2.131.6 pin\_center\_definition**

The `pin_center_definition` specifies the centerline distance between the centers of the flask pins that match the cope and drag sections of the mould. See 4.3.190 for the application assertion.

#### **4.2.131.7 type\_of\_flask**

The `type_of_flask` specifies the kind of flask. See 4.3.189 for the application assertion.

NOTE The type of flask may be a trade name, may be wooden or non-standard.

### **4.2.132 Flaskless**

The `Flaskless` is a mould that may require a removable flask jacket slipped around the mould when the mould is filled or is a mould made from a `Mould_box` (see 4.2.199) where the mould is free standing after box is removed. The data associated with a `Flaskless` are the following:

- `description`;
- `mould_made_by`.

#### **4.2.132.1 description**

The `description` specifies a word or group of words that describe the flaskless mould and the use of weights when required.

EXAMPLE example are steel tub, wooden box, removable flask jacket or may describe how the cope and drag moulds are matched for filling after the mould box is removed.

#### **4.2.132.2 mould\_made\_by**

The `mould_made_by` specifies a mould box size. See 4.3.192 for the application assertion.



### 4.2.133 Gating\_design\_feature

A `Gating_design_feature` is a type of `Feature` (see 4.2.124) that defines a functional characteristic of the gating system. A `Gating_design_feature` is either a `Choke` (see 4.2.34), `Feeders` (see 4.2.125), `Filter` (see 4.2.126), `Ingate` (see 4.2.152), `Ingate_contacts` (see 4.2.151), `Risers` (see 4.2.273), `Riser_contact` (see 4.2.272), `Runner` (see 4.2.274), `Sprue` (see 4.2.310), `Vent` (see 4.2.336) or `Well` (see 4.2.338). The data associated with a `Gating_design_feature` are the following:

- `placement`.

#### 4.2.133.1 placement

The `placement` specifies the position and orientation of a `Gating_design_feature` relative to the base shape for a part. See 4.3.193 for the application assertion.

### 4.2.134 Gating\_system

The `Gating_system` is a network of channels connected to a funnel like shape at the top of the mould through which molten metal flows into the mould cavity to produce the cast part.

NOTE This application object is harmonized with the `gating_system` definition from glossary of metalcasting terms from AFS [2].

The data associated with a `Gating_system` are the following:

- `gating_components`.

#### 4.2.134.1 gating\_components

The `gating_components` are the individual items or sets of items that form the network of channels to make the `Gating_system`. See 4.3.194, 4.3.195, 4.3.196 and 4.3.197 for the application assertion.

### 4.2.135 General\_rib

A `General_rib` is a type of `Rib` (see 4.2.271) that the rib sides or the rib top is not planar. The data associated with a `General_rib` are the following:

- `walls_top_floor`;
- `wall_to_top_fillet`;
- `wall_to_floor_fillet`;
- `part_rib_fillet`.

#### **4.2.135.1 walls\_top\_floor**

The `walls_top_floor` specifies one or more planar elements or non-planar surface elements for a rib side, a rib top, or a rib floor. See 4.3.199 and 4.3.200 for the application assertion.

#### **4.2.135.2 wall\_to\_top\_fillet**

The `wall_to_top_fillet` specifies the dimensional value for the transitional radius between the rib side and the rib top. See 4.3.198 for the application assertion.

#### **4.2.135.3 wall\_to\_floor\_fillet**

The `wall_to_floor_fillet` specifies the dimensional value for the transitional fillet radius between the rib side and the rib floor. See 4.3.198 for the application assertion.

NOTE The rib floor is defined as the casting body shape opposite the rib top.

#### **4.2.135.4 part\_rib\_fillet**

The `part_rib_fillet` specifies the dimensional value for the radius of a rib end that does not attach to any surface of the cast part. See 4.3.198 for the application assertion.

NOTE A `part_rib_fillet` is not defined as a rib top.

### **4.2.136 Generic\_casting\_resource**

A `Generic_casting_resource` is an inventory item used to support machining operations. Each `Generic_manufacturing_resource` is either a `Generic_casting_resource`, `Resource_with_material` (see 4.2.268), or `Resource_with_representation` (see 4.2.269).

NOTE This application object is harmonized with the `generic_manufacturing_resource` from ISO 10303-240:2005, Clause 4.2.53.

EXAMPLE Durable products such as marking dies, and consumable products such as dye penetrant, coolants, and lubricants are `Generic_casting_resources`.

The data associated with a `Generic_casting_resource` are the following:

- description;
- name;
- quantity;
- units.

### 4.2.136.1 description

The description specifies the word, or group of words, that describe the casting resource needed.

NOTE This description is required for human interpretation to distinguish it from other resources of the same type or family, such as hydraulic oil or lubricating oil.

### 4.2.136.2 name

The name specifies the word, or group of words, that make up the unique designation of a casting resource.

### 4.2.136.3 quantity

The quantity specifies the count of the Generic\_casting\_resource needed.

EXAMPLE Quantity such as 'three' in "three gallons", 'two' in "two pair", or 'one' in "one each".

### 4.2.136.4 units

The units specifies the physical measure of the Generic\_casting\_resource needed.

EXAMPLE Units such as 'gallons' in "three gallons", 'pair' as in "two pair", or 'each' as in "one each".

## 4.2.137 Geometric\_model

A Geometric\_model is a model containing complete representation of shape.

## 4.2.138 Geometric\_tolerance

A Geometric\_tolerance is the maximum or minimum variation from true geometric form or position that may be permitted in manufacture. Geometric\_tolerance should be employed only for those requirements of a part critical to its functioning or interchangeability. Each Geometric\_tolerance is one of the following: Angularity\_tolerance, Circular\_runout\_tolerance, Circularity\_tolerance, Concentricity\_tolerance, Cylindricity\_tolerance, Flatness\_tolerance, Linear\_profile\_tolerance, Parallelism\_tolerance, Perpendicularity\_tolerance, Position\_tolerance, Straightness\_tolerance, Surface\_profile\_tolerance, Symmetry\_tolerance, or a Total\_runout\_tolerance.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.104.

NOTE 2 Geometric tolerance definitions are derived from ISO 1101.

ISO 10303-223:2008 (E)

The data associated with a Geometric\_tolerance are the following:

- applied\_shape;
- geometric\_tolerance\_value;
- modifier\_control;
- significant\_digits;
- unit\_of\_measure;
- zone\_definition.

#### **4.2.138.1 applied\_shape**

The applied\_shape specifies the shape on a Part that is being tolerated by a Geometric\_tolerance. There may be more than one applied\_shape for a Geometric\_tolerance. See 4.3.202 for the application assertion.

#### **4.2.138.2 geometric\_tolerance\_value**

The geometric\_tolerance\_value specifies the tolerance amount that a part is allowed to meet the required accuracy for proper fit.

#### **4.2.138.3 modifier\_control**

The modifier\_control specifies the material condition which is applied to the shape being tolerated by the Geometric\_tolerance. The modifier\_control need not be specified for a particular Geometric\_tolerance. See 4.3.201 for the application assertion.

#### **4.2.138.4 significant\_digits**

The significant\_digits specifies the number of decimal places indicating the accuracy of the tolerance.

#### **4.2.138.5 unit\_of\_measure**

The unit\_of\_measure specifies the quantity of measure in which the value is given.

#### **4.2.138.6 zone\_definition**

The zone\_definition specifies the tolerance zone that restricts the Geometric\_tolerance. The zone\_definition need not be specified for a particular Geometric\_tolerance. See 4.3.203 for the application assertion.

### 4.2.139 Geometric\_tolerance\_precedence\_relationship

A `Geometric_tolerance_precedence_relationship` is a composite geometric tolerance. A part may have a shape that has a geometric tolerance and that shape is the basic shape for a pattern which may also have a geometric tolerance. The geometric tolerance for the base shape shall have precedence over the geometric tolerance for the pattern.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.105.

The data associated with a `Geometric_tolerance_precedence_relationship` are the following:

- `base_shape_tolerance`;
- `pattern_shape_tolerance`.

#### 4.2.139.1 base\_shape\_tolerance

The `base_shape_tolerance` specifies the `Geometric_tolerance` which is applied to a shape on a Part. See 4.3.204 for the application assertion.

#### 4.2.139.2 pattern\_shape\_tolerance

The `pattern_shape_tolerance` specifies the `Geometric_tolerance` which is applied to a pattern on a Part. See 4.3.204 for the application assertion.

### 4.2.140 Geometry\_characteristic

A `Geometry_characteristic` is the size, shape and geometric orientation of a `Material_structure_element` (see 4.2.189). The data associated with a `Geometry_characteristic` are the following:

- `characteristic_size`;
- `orientation`;
- `shape`.

#### 4.2.140.1 characteristic\_size

The `characteristic_size` specifies the dimensional measure used to describe the physical magnitude of a `Material_structure_element` (see 4.2.189).

EXAMPLE Particle diameter and plate length are dimensional measures.

The `characteristic_size` need not be specified for a particular `Geometry_characteristic`. See 4.3.206 for the application assertion.

### **4.2.140.2 orientation**

The orientation specifies the spatial placement of a `Material_structure_element` (see 4.2.189).

EXAMPLE The grain orientation in single crystal jet engine fan blades.

The orientation need not be specified for a particular `Geometry_characteristic`. See 4.3.205 for the application assertion.

### **4.2.140.3 shape**

The shape specifies the typical spatial form of a `Material_structure_element` (see 4.2.189). The shape need not be specified for a particular `Geometry_characteristic`. See 4.3.207 for the application assertion.

### **4.2.141 Hardcopy**

A Hardcopy is a type of `Document_file` (see 4.2.95) that is the actual stack of paper consisting of one or more sheets, on which some product data is written, printed or plotted.

### **4.2.142 Hardness**

A Hardness is the resistance of a material to deformation by external forces.

The data associated with a Hardness are the following:

- `high_value`;
- `low_value`;
- `nominal`;
- `scale`.

#### **4.2.142.1 high\_value**

The `high_value` specifies the highest allowed value of hardness for a specific material type. The `high_value` need not be specified for a particular Hardness.

#### **4.2.142.2 low\_value**

The `low_value` specifies the lowest allowed value of hardness for a specific material type. The `low_value` need not be specified for a particular Hardness.

### 4.2.142.3 nominal

The nominal specifies the nominal value of hardness for a specific material type.

### 4.2.142.4 scale

The scale specifies the method of determining hardness.

EXAMPLE Rockwell and Brinell are examples of scale.

## 4.2.143 Heat

A Heat is a batch of metal, obtained from an uninterrupted melting operation in an individual cupola or furnace. The data associated with a Heat are the following:

- heat\_number;
- production\_date.

### 4.2.143.1 heat\_number

The heat number specifies the label used to identify a Heat.

### 4.2.143.2 production\_date

The production date specifies the date and time when the melting of a Heat was completed. See 4.3.208 for the application assertion.

## 4.2.144 Heat\_treat

In Heat\_treat is a type of Non\_casting\_operations (see 4.2.202) that identifies the activity of heat treating the cast part. The data associated with a Heat\_treat are the following:

- requirement;

### 4.2.144.1 requirement

The requirement specifies the mechanical properties that shall be met for the physical properties of the material used for the cast part. See 4.3.209 for the application assertion.

## **4.2.145 Heat\_treat\_requirememnt**

A Heat\_treat\_requirement is a type of Process\_requirement (see 4.2.240) that is a sequence of heating and cooling cycles used to obtain desired properties or conditions of an Cast\_part (see 4.2.19). The data associated with a Heat\_treat\_requirement are the following:

- additional\_requirements;
- heat\_treat\_specificatiion;
- heat\_treatment\_per\_design\_data;
- mechanical\_property;
- softening\_anneal\_required;
- stress\_relief\_required;
- where\_heat\_treat\_is\_to\_be\_performed;

### **4.2.145.1 additional\_requirements**

The additional\_requirements specifies specific conditions and criteria that are not defined or available in a standard or publicly available specification. The additional\_requirements need not be specified for a particular Heat\_treat\_requirement for the cast part.

### **4.2.145.2 heat\_treat\_specificatiion**

The heat\_treat\_specification specifies the document or standard where the specification requirements for the heat treatment are defined. The heat\_treat\_specification need not be specified for a particular Heat\_treat\_requirement for the cast part. See 4.3.211 for the application assertion.

### **4.2.145.3 mechanical\_property**

The mechanical\_property specifies one or more elastic and inelastic properties for which a cast part is designed. See 4.3.210 for the application assertion.

### **4.2.145.4 heat\_treatment\_per\_design\_data**

The heat\_treatment\_per\_design\_data specifies that the heat treat process property characteristics are defined by the design drawing or design record. See 4.3.212 for the application assertion.



#### 4.2.145.5 softening\_anneal\_required

The `softening_anneal_required` specifies a Boolean value that indicates if the customer requires annealing for the casting. A value of true specifies that softening anneal is required. A value of false specifies that softening anneal is not required.

#### 4.2.145.6 stress\_relief\_required

The `stress_relief_required` specifies a Boolean value that indicates if the customer requires stress relief for the casting. A value of true specifies that stress relief is required. A value of false specifies that stress relief is not required.

#### 4.2.145.7 where\_heat\_treat\_is\_to\_be\_performed

The `where_heat_treat_is_performed` specifies who and where the heat treatment is done.

NOTE heat treatment could be done at the foundry, by the customer, or at a heat treat facility. If a heat treat facility is to do the heat treating then the facility name and address are to be stated in the text string.

### 4.2.146 Height\_dimension

A `Height_dimension` is a type of `Size_tolerance` (see 4.2.304) that specifies the size along a straight line that is referred to as height in the referenced shape.

NOTE This application object is harmonized with the same from ISO 10303-224:2006, Clause 4.2.108.

The data associated with a `Height_dimension` are the following:

— `path`.

#### 4.2.146.1 path

The `path` specifies the shape that the tolerance applies to. A `Height_dimension` may but need not require a `path`. See 4.3.213 for the application assertion.

### **4.2.147 Illustration**

An Illustration is a pictorial presentation used for clarification of an Part\_version (see 4.2.217) where the narrative alone would not suffice.

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.59.

The data associated with an Illustration are the following:

- description;
- id;
- is\_owned\_by.

#### **4.2.147.1 description**

The description specifies the word, or group of words, that describe the purpose of the Illustration. A description need not be specified for a particular Illustration.

#### **4.2.147.2 id**

The id specifies the unique identification, within an organization, of an Illustration.

#### **4.2.147.3 is\_owned\_by**

The is\_owned\_by specifies the View\_reference (see 4.2.337) containing the illustration. A View\_reference need not be specified for a particular Illustration. See 4.3.214 for the application assertion.

### **4.2.148 Impregnation\_activity**

In Impregnation\_activity is a type of Non\_casting\_operations (see 4.2.202) that identifies the activity of sealing castings, ferrous and nonferrous, where micro porosity is present.

### **4.2.149 In\_facility\_location**

An In\_facility\_location is the geographic position, within the enterprise, where a work cell is to be found.

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.61.

EXAMPLE The Work\_cell is located at the hi-bay door, column SW5, building 88. The building\_or\_area is building 88, the location\_code is column SW5, and the sub location is the hi-bay door.

The data associated with an `In_facility_location` are the following:

- `building_or_area`;
- `location_code`;
- `sublocation`.

#### **4.2.149.1 building\_or\_area**

The `building_or_area` specifies the unique identification, within an organization, of a structure or place.

#### **4.2.149.2 location\_code**

The `location_code` specifies the unique identification, within an organization, of a geographic position within a building.

#### **4.2.149.3 sublocation**

The `sublocation` specifies the unique identification, within an organization, of a further refinement of the `In_facility_location`.

#### **4.2.150 Indirect\_stock\_requisition**

An `Indirect_stock_requisition` is a type of `Requisition` (see 4.2.265) that defines indirect items required to support part manufacturing.

NOTE This application object is harmonized with the same from ISO 10303-224:2006, Clause 4.2.113.

#### **4.2.151 Ingate\_contacts**

The `Ingate_contacts` is a type of `Gating_design_feature` (see 4.2.133) that is the connection information for each ingate at the point of contact with the production pattern and with the runner bar.

NOTE there will be two contacts for each end ingate. One at the production pattern and one at the runner.

The data associated with an `Ingate_contacts` are the following:

- `applied_to`;
- `contact_placement`;
- `break_off_connection`;
- `pattern_transition`;

ISO 10303-223:2008 (E)

- runner\_connection;
- runner\_transition;
- shape.

#### **4.2.151.1 applied\_to**

An applied\_to specifies the Production\_pattern to which the ingate\_contacts are valid. See 4.3.215 for the application assertion.

#### **4.2.151.2 contact\_placement**

The contact\_placement specifies whether the Ingate\_contact is located in the cope side or in the drag side of the mould.

NOTE See 4.2.151.2.1 to 4.2.151.2.3 for the definition of each allowable value for contact\_placement.

**4.2.151.2.1 cope:** the ingate is located in the cope side of the mould.

**4.2.151.2.2 drag:** the ingate is located in the drag side of the mould.

**4.2.151.2.3 cope\_and\_drag:** the ingate is located in both the cope and drag side of the mould.

#### **4.2.151.3 break\_off\_connection**

The break\_off\_connection specifies the geometric shape and dimensional information. See 4.3.216 for the application assertion.

NOTE A break-off is located at the casting as part of an ingate or a feeder from the riser. This shape applies only to the break off shape not to the ingate contact in general.

#### **4.2.151.4 pattern\_transition**

The pattern\_transition specifies the edge\_round and fillets where each ingate makes contact with the Production\_pattern. See 4.3.217 for the application assertion.

#### **4.2.151.5 runner\_connection**

The runner\_connection specifies the geometric shape and dimensional information from the ingate to the runner connection. See 4.3.217 for the application assertion.

NOTE Where the ingate make connect with the runner all sharp edges should be eliminated with fillets and edge round transitions to provide for a smooth free flow of molten metal from the runner into the ingate.

#### 4.2.151.6 runner\_transition

The runner\_transition specifies the edge\_round and fillets where each ingate makes contact with the Runner. See 4.3.217 for the application assertion.

#### 4.2.152 Ingate

The Ingate is a type of Gating\_design\_feature (see 4.2.133) that are the individual channels that connect the mould cavity to a runner through which molten metal flows.

NOTE 1 The Ingate is the smallest of the channels that molten metal flows.

NOTE 2 This application object is harmonized with the gate object from ISO 8062-1:2007, Clause 3.5.

The data associated with an Ingate are the following:

- contacts;
- cope\_or\_drag;
- parametric\_ingate\_shape.

##### 4.2.152.1 contacts

The contacts specifies the form of each ingate connection to the pattern through which the molten metal enters the mould cavity. See 4.3.219 for the application assertion.

##### 4.2.152.2 cope\_or\_drag

The cope\_or\_drag specifies the placement of each ingate. Each will be located in either the cope or in the drag of the mould.

NOTE See 4.2.152.2.1 to 4.2.152.2.3 for the definition of each allowable value for contact\_placement.

**4.2.152.2.1 cope:** the ingate is located in the cope side of the mould.

NOTE This application object is harmonized with the cope definition from glossary of metalcasting terms from AFS [2].

**4.2.152.2.2 drag:** the ingate is located in the drag side of the mould.

NOTE This application object is harmonized with the drag definition from glossary of metalcasting terms from AFS [2].

**4.2.152.2.3 cope\_and\_drag:** the ingate is located in both the cope and drag side of the mould.

### **4.2.152.3 parametric\_ingate\_shape**

The `parametric_ingate_dimensions` specifies the size of each ingate channel from the Production pattern to the Runner. See 4.3.220 for the application assertion.

### **4.2.153 Inspection\_activity**

An `Inspection_activity` is a type of `Casting_activity` (see 4.2.21) that is an action, procedure, or activity used to evaluate, check, inspect, or validate a cast part or tooling used to produce a cast part. Each `Inspection_activity` may be one of the following: `Dimensional_measurement_inspection` (see 4.2.90) or `Bench_inspection` (see 4.2.16). The data associated with an `Inspection_activity` are the following:

- `customer_requirements`;
- `foundry_approval`;
- `inspected_part`;
- `inspection_level`;
- `inspection_report`;
- `quality_assurance`;
- `reason_for_inspection`;
- `type_of_inspection`.

#### **4.2.153.1 customer\_requirements**

The `customer_requirements` specifies a predetermined set of procedures for inspecting and testing a cast part or the tooling for a cast part and corrective actions when failure occurs. See 4.3.223 for the application assertion.

#### **4.2.153.2 foundry\_approval**

The `foundry_approval` specifies the individual in the foundry organization or their agent by person and organization that inspected the casting or the tooling that produced the approved castings. See 4.3.221 for the application assertion.

#### **4.2.153.3 inspected\_part**

The `inspected_part` specifies the cast part version to be or has been examined, checked, tested, or verified. See 4.3.224 for the application assertion.

#### 4.2.153.4 inspection\_level

The `inspection_level` specifies an integer from 1 to 5 that specifies the documentation level and records required for submission to the customer for a Production Part Approval Process (PPAP). The `inspection_level` need not be specified for a particular `Inspection_activity`.

#### 4.2.153.5 inspection\_report

An `inspection_report` specifies the failure results of the action, procedure, or activity used to evaluate, check, inspect, or validate a cast part or tooling used to produce a cast part. See 4.3.223 for the application assertion.

#### 4.2.153.6 quality\_assurance

The `quality_assurance` specifies the process or procedures necessary to determine and assure that the design requirements for the quality of the cast part are met. See 4.3.222 for the application assertion.

#### 4.2.153.7 reason\_for\_inspection

The `reason_for_inspection` specifies the justification for the examination of the cast part or the cast part tooling as required by the customer.

NOTE See 4.2.153.7.1 to 4.2.153.7.8 for the definition of each allowable value of `reason_for_inspection`.

**4.2.153.7.1 initial inspection:** is a first article inspection or a required PPAP inspection of the cast part.

**4.2.153.7.2 engineering change:** the cast part is inspected following a completed engineering change order that required some tooling modification or some value added operation.

**4.2.153.7.3 tooling replacement:** a worn tool is replaced with a new tool, inspection of the cast part is required.

**4.2.153.7.4 tooling transfer:** a tool that has been moved from one casting source to another casting source, inspection of the cast part is required.

**4.2.153.7.5 tooling refurbishment:** a tool that has wear and has been repaired, reconditioned, or overhauled inspection of the cast part is required.

**4.2.153.7.6 discrepancy correction:** inspection due to tooling or process procedure problems in producing the cast part

**4.2.153.7.7 material or supplier source changed:** cast part is inspected due to changes in materials or a different supplier.

**4.2.153.7.8 production sample set:** inspection of a selected or pre-determined group of cast parts.

### 4.2.153.8 type\_of\_inspection

The type\_of\_inspection specifies the required examination of the cast part or the cast part tooling to the design specification.

NOTE See 4.2.153.8.1 to 4.2.153.8.10 for the definition of each allowable value for inspection\_type.

**4.2.153.8.1 destructive:** the cast part is destroyed, permanently damaged or altered during the inspection activity or procedure.

**4.2.153.8.2 dimensional:** each non-reference dimension is measured for accuracy against the design specification.

**4.2.153.8.3 functional\_testing:** the cast part is inspected for functional or performance requirements.

**4.2.153.8.4 gauging:** checking of the cast part is performed by use of instruments, jigs, or fixtures.

EXAMPLE go no-go gauges etc.

**4.2.153.8.5 hardness\_property:** the cast part is tested for the required hardness as per the design record.

**4.2.153.8.6 internal\_inspection:** the metal of the cast part is tested for integrity and soundness.

EXAMPLE no voids, porosity, checks, etc

**4.2.153.8.7 leak testing:** the casting checked using pressure

**4.2.153.8.8 material composition:** the metal used for the cast part is tested for its proper metallic properties and the percentage of each specific chemical element.

**4.2.153.8.9 surface:** the exterior of the cast part is inspected for correct surface conditions as per the design record.

EXAMPLE: blow hole, cold shut, mis-run, scabing, severe core shift are examples of casting defects.

**4.2.153.8.10 visual:** an inspection using only the human eye as the inspection tool.

NOTE most times visual inspection is the first level of inspection.

### 4.2.154 Inspection\_or\_test\_requirement

An Inspection\_or\_test\_requirement is a type of Casting\_requirement (see 4.2.62) that is the examination procedure to check a Part\_version (see 4.2.217) for compliance with a desired condition.



The data associated with an `Inspection_or_test_requirement` are the following:

- `frequency`;
- `inspection_against`;
- `production_quality_requirement_standard`;
- `sampling_run_size`;
- `test_name`;
- `validation_id`.

#### **4.2.154.1 frequency**

The `frequency` specifies the how often the procedure shall be performed. See 4.3.227 for the application assertion.

#### **4.2.154.2 inspection\_against**

The `inspection_against` specifies the requirements for which the procedure verifies compliance. See 4.3.226 for the application assertion.

#### **4.2.154.3 production\_quality\_requirement\_standard**

The `production_quality_requirement_standard` specifies one or more standards required by the customer for evaluating the quality of a cast part for customer acceptance. See 4.3.229 for the application assertion.

NOTE in addition to standard approved by a Standards Body, i.e., ISO or ANSI a customer may require an industry standard or an internal standard developed by the customer.

#### **4.2.154.4 sampling\_run\_size**

The `sampling_run_size` specifies the number of castings that shall be produced and tested in the sample run prior to production. The sampling run size need not be specified for a particular `Inspection_or_test_requirement`. See 4.3.227 for the application assertion.

#### **4.2.154.5 test\_name**

The `test_name` specifies the label used to identify the `Inspection_or_test_requirement`. The `test_name` need not be specified for a particular `Inspection_or_test_requirement`. See 4.3.228 for the application assertion.

#### **4.2.154.6 validation\_id**

The `validation_id` specifies a unique identification for the inspection or test required by the `Inspection_plan` (see 4.2.156).

#### **4.2.155 Inspection\_or\_test\_result**

An `Inspection_or_test_result` is the outcome of an examination or measurement. Each `Inspection_or_test_result` may be a `Composition_inspection` (see 4.2.39), `Property_inspection` (see 4.2.255), `Other_inspection` (see 4.2.155), or a `Tolerance_inspection` (see 4.2.324). The data associated with an `Inspection_or_test_result` are the following:

- `aid_used`;
- `applies_to`;
- `belongs_to`;
- `description`;
- `result`.

##### **4.2.155.1 aid\_used**

An `aid_used` specifies one or more devices employed during the inspection or test procedures. See 4.3.231 for the application assertion.

##### **4.2.155.2 applies\_to**

The `applies_to` specifies the location on an item where the inspection was made. The `applies_to` need not be specified for a particular `Inspection_or_test_result`. See 4.3.232 for the application assertion.

NOTE Cast parts may be examined for porosity or cracks along a specified surface, where these defects would be problematical.

##### **4.2.155.3 belongs\_to**

The `belongs_to` specifies the `Sampled_set` (see 4.2.275) that was inspected. See 4.3.231 for the application assertion.

##### **4.2.155.4 description**

The `description` specifies the textual account of the `Inspection_or_test_result`.

NOTE The `description` may contain an account of porosity or other defects found in a casting, and the results of any mechanical tests made.

#### 4.2.155.5 result

The result specifies the outcome of the Inspection\_or\_test\_result.

#### 4.2.156 Inspection\_plan

An Inspection\_plan is a scheme of predetermined processes or procedures employed to evaluate the quality and dimensional accuracy of a cast part against the design record. The data associated with an Inspection\_plan are the following:

- correction\_actions;
- correction\_description;
- correction\_procedures;
- customer\_requirements.

##### 4.2.156.1 corrective\_action

The corrective\_action specifies a permitted remedy to accept a cast part for delivery.

NOTE See 4.2.156.1.1 to 4.2.156.1.5 for the definition of each allowable value for correction\_action\_types.

**4.2.156.1.1 coating:** a process of applying a liquid material to a casting

**4.2.156.1.2 grinding:** a process for removing unexpected material from a casting.

**4.2.156.1.3 hipping:** a Hot Isostatic Pressing process used to improve internal soundness or integrity, increase density, and improve mechanical or fatigue properties of a casting.

**4.2.156.1.4 impregnation:** a process for accepting a leaky casting.

**4.2.156.1.5 welding:** a process for preserving a casting by adding additional material to the casting.

##### 4.2.156.2 correction\_description

A correction\_description is a textual explanation or narrative of under what conditions a correction\_procedure and correction\_action may be employed and what materials may be used to salvage a cast part.

### **4.2.156.3 correction\_procedures**

An `correction_procedure` specifies what inspection process may have a permitted remedy for accepting a cast part.

NOTE 1 A customer may permit a specific remedy for correcting a specific area, section, or appearance of the cast part that failed during one of the inspection procedures.

NOTE 2 See 4.2.156.3.1 to 4.2.156.3.3 for the definition of each allowable value for `correction_procedures_types`.

**4.2.156.3.1 dimensional:** a permitted remedy to correct a dimensional failure.

**4.2.156.3.2 visual:** a permitted remedy to correct a visual failure.

EXAMPLE visual failure could be an area of the casting that has surface imperfections that could be welded or additional grinding to correct the affected area.

**4.2.156.3.3 internal:** a permitted remedy to correct an internal failure.

EXAMPLE internal failure could be a leaker that could be corrected through impregnation or welding

### **4.2.156.4 customer\_requirements**

The `customer_requirements` specifies one or more customer requirements for inspection and testing. See 4.3.233 and 4.3.234 for the application assertion.

### **4.2.157 Integration\_interval**

An `Integration_interval` is a time step employed for numerical integration, and the total time period covered by a portion of the overall simulation run. The data associated with an `Integration_interval` are the following:

- `delta_time`;
- `length_of_elapsed_time_interval`.

#### **4.2.157.1 delta\_time**

The `delta_time` specifies the time step employed for a numerical integration step. See 4.3.235 for the application assertion.

#### **4.2.157.2 length\_of\_elapsed\_time\_interval**

The `length_of_elapsed_time_interval` specifies the total time period covered by the `Integration_interval`. See 4.3.235 for the application assertion.

### 4.2.158 Investment\_casting\_master

An `Investment_casting_master` is a type of `Investment_casting_tooling` (see 4.2.159) that is a scale replica of the production tooling needed to produce cast part. The scale factor or shrink multiplier will be determined by the material and the process used to produce the production tooling. The data associated with an `Investment_casting_master` are the following:

- `shrink_factor_for_master_tooling`

#### 4.2.158.1 shrink\_factor\_for\_master\_tooling

The `shrink_factor_for_master_tooling` specifies specifies the patternmaker's rule that compensates for double shrinkage. One shrinkage is required for the tooling casting shrinking in size as it cools to ambient temperature and the other shrinkage is required for the production tooling as it comes to ambient temperature.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE 1 if the cast part has a shrink factor of 1/8 inches and the production tooling has a shrink factor of 1/8 inches the master shrink factor is 1/4 inches or double shrinkage.

EXAMPLE 2 if the shrinkage compensation is one quarter inch per foot the shrink rule will be one quarter of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0208333 inches per inch or where the foot shrink rule will be 12.25 longer than the standard 12 inch foot rule.

### 4.2.159 Investment\_casting\_tooling

The `Investment_casting_tooling` are the reusable metal moulds and components used to form investment patterns and cores to produce precision near\_net shape investment cast parts. The data associated with an `Investment_casting_tooling` are the following:

- `investment_tooling_or_prototype`;
- `material_definition`;
- `tooling_identifier`.

#### 4.2.159.1 investment\_tooling\_or\_prototype

The `investment_tooling_or_prototype` specifies the method used to produce the investment pattern and gating system. The tooling is a metal tool or the prototype is a rapid prototyping material using a STL file to create the pattern. See 4.3.237 and 4.3.236 for the application assertion.

### **4.2.159.2 material\_definition**

The `material_definition` specifies a primary material choice for producing the tooling or the prototype pattern. There may be more than one `material_definition` for an `Investment_casting_tooling`.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for `core_box_material` type

### **4.2.159.3 tooling\_identifier**

The `tooling_identifier` specifies an alphanumeric string that identifies the investment tooling used to produce the cast part.

### **4.2.160 Investment\_casting\_tooling\_definition**

An `Investment_casting_tooling_definition` defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces. An `Investment_casting_tooling_definition` may first be the definition for an `Investment_casting_master` that will become the production tooling. The `Investment_casting_tooling_definition` may be an `Investment_casting_master` (see 4.2.158).

The data associated with an `Investment_casting_tooling_definition` are the following:

- `core_tooling`;
- `metal_flow_system`;
- `pattern_section_assembly`;
- `pattern_section_assemblt`;
- `produces`;
- `tree_structure`.

#### **4.2.160.1 core\_tooling**

The `core_tooling` specifies the requirements for one or more metal moulds that form the core(s) needed for the investment casting. The `core_tooling` need not be specified for a particular `Investment_casting_tooling_definition`. See 4.3.240 for the application assertion.

#### **4.2.160.2 metal\_flow\_system**

The `metal_flow_system` specifies the sprue and runner(s) attached to the investment mould. See 4.3.244 for the application assertion.

### 4.2.160.3 **pattern\_section\_assembly**

The `pattern_section_assembly` specifies how the sections produced are to be assembled to form the complete pattern. See 4.3.241 for the application assertion.

### 4.2.160.4 **pattern\_section\_definition**

The `pattern_section_definition` specifies the tool cavity shape for the pattern section to be formed by the tool. See 4.3.242 for the application assertion.

### 4.2.160.5 **produces**

The `produces` specifies the type of mould or die that is created by the investment casting tooling. See 4.3.238 and 4.3.239 for the application assertion.

### 4.2.160.6 **tree\_structure**

The `tree_structure` specifies the configuration of the `Investment_mould` (see 4.2.162) with the sprue and runners of a gating system. See 4.3.243 for the application assertion.

## 4.2.161 **Investment\_design\_feature**

An `Investment_design_feature` is a type of `Feature` (see 4.2.124) that is required in the design of an investment casting mould. The `Investment_design_feature` is either an `Investment_mould` (see 4.2.162) or a `Sprue_and_runner_mould` (see 4.2.311). The data associated with a `Investment_design_feature` are the following:

- placement.

### 4.2.161.1 **placement**

The `placement` specifies the position and orientation of a `Investment_design_feature` relative to the base shape for a part. See 4.3.245 for the application assertion.

## 4.2.162 **Investment\_mould**

An `Investment_mould` is a type of `Investment_design_feature` (see 4.2.161) that is the design of a mould constructed of refractory material around wax or other expendable patterns used for investment casting.

EXAMPLE Investment moulds may be made of wax, plastic, lead.

NOTE The `Investment_mould` includes a description of the mould shape or thickness, and a description of each layer in the case of shell moulds.

ISO 10303-223:2008 (E)

The data associated with an `Investment_mould` are the following:

- `mould_material`;
- `number_of_pattern`.

#### **4.2.162.1 mould\_material**

The `mould_material` specifies the word or group of words that describe or identify the substance for the `Investment_mould`.

EXAMPLE the substance for the mould could be ceramic, aluminum, steel, etc.

#### **4.2.162.2 number\_of\_pattern**

The `numbers_of_patterns` specifies the total number of patterns attached to the mould cavity tree.

#### **4.2.163 Item\_size**

The `Item_size` specifies a dimensional value with the appropriate units assigned to a particular item or detail. The data associated with an `Item_size` are the following:

- `item_height`
- `item_width`;
- `item_length`;
- `item_draft`;
- `item_diameter`.

##### **4.2.163.1 item\_height**

The `item_height` is the measured perpendicular distance with appropriate units from the base of an item to its top. See 4.3.246 for the application assertion.

NOTE sometimes height and thickness are used interchangeably for an item.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 2 inch value is the `item_height`.



### 4.2.163.2 item\_width

The `item_width` is the measured perpendicular distance with appropriate units from one side of an item to its side opposite. See 4.3.246 for the application assertion.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 8 inch value is the `item_width`.

### 4.2.163.3 item\_length

The `item_length` is the measured perpendicular distance with appropriate units from one end of an item to its end opposite. See 4.3.246 for the application assertion.

EXAMPLE A block of steel that measures 2 inches by 8 inches by 3 feet in size. The 3 feet value is the `item_length`.

### 4.2.163.4 item\_draft

The `item_draft` is the measured value with units of the slant or taper required to remove an item from the mould. See 4.3.246 for the application assertion.

NOTE the slant or taper is most often defined as a measured angle with units in degrees.

### 4.2.163.5 item\_diameter

The `item_diameter` is the measured distance with appropriate units defined as 2 times the radius value that has the same units. See 4.3.246 for the application assertion.

NOTE The `item_diameter` may also be defined as the measured distance with appropriate units of a straight line that has two points on the circumference of a circle or sphere such that the straight line passes through the centre of the circle or sphere.

EXAMPLE A round bar of steel that measures 2 inches by 3 feet in size. The 2 inch value is the `item_diameter`.

## 4.2.164 Length\_dimension

A `Length_dimension` is a type of `Size_tolerance` (see 4.2.304) that specifies the size along a straight line that is referred to as length in the referenced shape.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.115.

The data associated with a `Length_dimension` are the following:

— `path`.

### 4.2.164.1 path

The `path` specifies the shape that the tolerance applies to. A `Length_dimension` may but need not require a `path`. See 4.3.247 for the application assertion.

## **4.2.165 Library\_part\_assignment**

An `Library_part_assignment` provides the means to reference information about a class within a parts library dictionary. The data associated with an `Library_part_assignment` are the following:

- `definitional_class_BSU`;
- `definitional_property_value_pairs`.

### **4.2.165.1 definitional\_class\_BSU**

The `definitional_class_BSU` specifies the identification of the component within a parts library as defined by ISO-13584. See 4.3.249 for the application assertion.

### **4.2.165.2 definitional\_property\_value\_pairs**

The `definitional_property_value_pairs` specifies the set of pairs (`Property_BSU` (see 4.2.254), `Property_value` (see 4.2.258)) defining the properties of the class. See 4.3.248 for the application assertion.

## **4.2.166 Limits\_and\_fits**

A `Limits_and_fits` contains the necessary information to express a tolerance of the limits-and-fits system standardized by ISO 286-1 and ISO 286-2.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.117.

The data associated with a `Limits_and_fits` are the following:

- deviation;
- fitting type;
- grade.

### **4.2.166.1 deviation**

The deviation specifies the class descriptor, by characters, for the designated limits and fits.

NOTE The characters 'A' to 'ZC' for holes or 'a' to 'zc' for shafts may be used for deviation.

### **4.2.166.2 fitting type**

The fitting type specifies whether the tolerance declaration applies to a shaft or to a hole. The fitting type need not be specified for a particular `Limits_and_fits`.

### 4.2.166.3 grade

The grade specifies the quality or the accuracy grade of a tolerance.

NOTE The grade is based on the international standard tolerance grade IT01 to IT18.

### 4.2.167 Location\_dimension\_tolerance

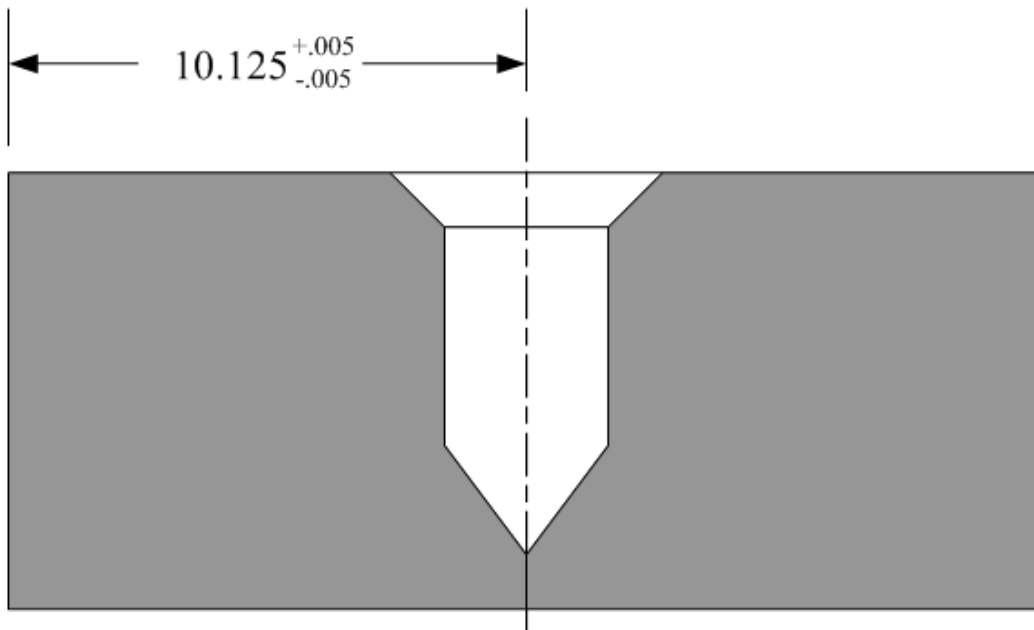
A Location\_dimension\_tolerance is a type of Location\_tolerance (see 4.2.169) that is the allowable variation in locating one feature of a part with respect to another.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.121.

NOTE 2 Figure 14 illustrates the Location\_dimension\_tolerance used to locate the center of a hole to the side of the part.

The data associated with a Location\_dimension\_tolerance are the following:

— plane\_and\_direction.



**Figure 14 — Location\_dimension\_tolerance**

### **4.2.167.1 plane\_and\_direction**

The `plane_and_direction` specifies a plane that contains the geometry for the `Location_dimension_tolerance` and a direction that is the location of the plane that contains the `Location_dimension_tolerance`. The `plane_and_direction` need not be specified for a particular `Location_dimension_tolerance`. See 4.3.250 for the application assertion.

EXAMPLE A part might be viewed in a front view for defining a `location_dimension_tolerance`.

### **4.2.168 Location\_element**

A `Location_element` is a type of `Shape_element` (see 4.2.285) that is a `Shape_aspect` representation for a reference point.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.122.

### **4.2.169 Location\_tolerance**

A `Location_tolerance` is a type of `Dimensional_tolerance` (see 4.2.91) that defines tolerances that are an allowable variation in location between an origin shape and a termination shape. Each `Location_tolerance` is either an `Angular_dimensional_tolerance` (see 4.2.10), `Location_dimension_tolerance` (see 4.2.167), or `Distance_along_curve_tolerance` (see 4.2.93).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.123.

The data associated with a `Location_tolerance` are the following:

- `directed`;
- `origin_shape`;
- `termination_shape`.

#### **4.2.169.1 directed**

The `directed` specifies a logical value designating the importance of direction for measuring a `location_dimension_tolerance`. If value is `TRUE`, `location_dimension_tolerance` is measured from point of origin to point of termination, if `FALSE`, an in tolerance result shall occur regardless of direction of measurement.

#### **4.2.169.2 origin\_shape**

The `origin_shape` specifies the shape on the Part that defines the starting position for a `Location_tolerance`. See 4.3.251 for the application assertion.

### 4.2.169.3 termination\_shape

The `termination_shape` specifies the shape on the Part that defines the ending position for a `Location_tolerance`. See 4.3.251 for the application assertion.

### 4.2.170 Lost\_foam\_casting\_die

A `Lost_foam_casting_die` is a type of `Production_tool` (see 4.2.246) that is the tooling to form the pattern, pattern sections, or gating features to make the lost foam pattern. The data associated with a `Lost_foam_casting_die` are the following:

- `defines_flask_type`;
- `die_wall_thickness`;
- `fill_gun_location`;
- `fill_gun_size`;
- `gating_requirements`;
- `mould_cooling`;
- `movable_die_item`;
- `pattern_removal_instruction`;
- `raw_bead_size`;
- `stationary_die`;
- `stationary_die_mounting`;
- `steam_line_connection`;
- `steam_line_location`;
- `tool_stored`.

#### 4.2.170.1 defines\_flask\_type

The `defines_flask_type` specifies the requirements for the flask to be used with the lost foam pattern. See 4.3.256 for the application assertion.

#### **4.2.170.2 die\_wall\_thickness**

The `die_wall_thickness` specifies the metal thickness of the die cavity that forms the foam pattern. There may be more than one `die_wall_thickness` for a `Lost_foam_casting_die`.

#### **4.2.170.3 fill\_gun\_location**

The `fill_gun_location` specifies the direction and location of the bead gun access for filling the die. There may be more than one `fill_gun_location` for a `Lost_foam_casting_die`. See 4.3.260 for the application assertion.

#### **4.2.170.4 fill\_gun\_size**

The `fill_gun_size` specifies the parameters of the blow area in the die for the bead fill gun attachment. There may be more than one `fill_gun_size` for a `Lost_foam_casting_die`. See 4.3.258 for the application assertion.

#### **4.2.170.5 gating\_requirements**

The `gating_requirements` specifies the features required in the `Gating_system` (see 4.2.134) for the lost foam casting. There may be more than one `gating_requirement` for a `Lost_foam_casting_die`. See 4.3.259 for the application assertion.

#### **4.2.170.6 mould\_cooling**

The `mould_cooling` specifies the inlet and outlet `Cooling_port` (see 4.2.47) requirements for the lost foam tool. There may be more than one `moulding_cooling` for a `Lost_foam_casting_die`. See 4.3.253 for the application assertion.

#### **4.2.170.7 movable\_die\_item**

The `movable_die_item` specifies the components of the die that move when the die mould is opened or closed. There may be more than one `movable_die_item` for a `Lost_foam_casting_die`. A `movable_die_item` need not be specified for a particular `Lost_foam_casting_die`. See 4.3.254 and 4.3.255 for the application assertion.

#### **4.2.170.8 pattern\_removal\_instruction**

The `pattern_removal_instruction` specifies the group of words that describe the process to remove the foam pattern or pattern section from the die. There may be more than one `pattern_removal_instruction` for a `Lost_foam_casting_die`. A `pattern_removal_instruction` need not be specified for a particular `Lost_foam_casting_die`.

**4.2.170.9 raw\_bead\_size**

The `raw_bead_size` specifies the size of the unexpanded polystyrene bead.

**4.2.170.10 stationary\_die**

The `stationary_die` specifies the die item(s) that is attached to a polystyrene bead blowing machine and does not move when the die mould is opened or closed. See 4.3.254 for the application assertion.

**4.2.170.11 stationary\_die\_mounting**

The `stationary_die_mounting` specifies mounting requirements to secure the lost foam die on the bead blowing machine or steam chest. See 4.3.257 for the application assertion.

**4.2.170.12 steam\_line\_connection**

The `steam_line_connection` specifies the input output steam line port to the die. There may be more than one `steam_line_connection` for a `Lost_foam_casting_die`.

**4.2.170.13 steam\_line\_location**

The `steam_line_location` specifies the direction and location of the steam line access holes to the die cavity. There may be more than one `steam_line_location` for a `Lost_foam_casting_die`. See 4.3.260 for the application assertion.

**4.2.170.14 tool\_stored**

The `tool_stored` specifies the place of storage for the `Lost_foam_casting_die`. See 4.3.261 and 4.3.262 for the application assertion.

**4.2.171 Loose\_piece**

A `Loose_piece` is a type of `Sand_casting_design_feature` (see 4.2.276) that is required to make one or more features that can not be moulded directly with the production pattern or the required core boxes. The loose piece shall be removed independently from the mould or core.

Note 1 The loose piece might be dovetailed to the pattern or pinned on with dowels. If by dowels, the sand is packed (rammed) tightly around the dowels and the loose piece to hold it in position. The dowels are withdrawn and the mould finished in the usual manner.

NOTE 2 This application object is harmonized with the loose piece definition from glossary of metalcasting terms from AFS [2].

ISO 10303-223:2008 (E)

The data associated with a Loose\_piece are the following:

- material\_type;
- shape;

#### **4.2.171.1 material\_type**

The material\_type specifies the substance used to fabricate the loose piece.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for \_material type

#### **4.2.171.2 shape**

The shape specifies the geometric design and form of the loose piece with details that define the dimensions with units. See 4.3.252 for the application assertion.

#### **4.2.172 Lot**

A Lot is a group of Master\_sample (see 4.2.182) objects that are manufactured together or in a sequence.

NOTE A lot may be established by the foundry; for example, to record the day when a group of castings are made. A lot may also be specified by the customer; for example, when a lot of 500 castings are specified for delivery monthly.

The data associated with a Lot are the following:

- lot\_number;
- production\_date.

#### **4.2.172.1 lot\_number**

The lot number specifies the numerical label used to identify the Lot.

#### **4.2.172.2 production\_date**

The production date specifies the date and time when the lot was manufactured, or when production of the lot was completed. See 4.3.263 for the application assertion.

#### **4.2.173 Machine\_mounting**

The Machine\_mounting is the method used to secure a Core\_box to a core machine or die tooling to a machine.

NOTE Most core machines produce cores as shell cores. Iso Cure cores, or CO2 cores. Each of these type core machines require different mounting requirements.



The data associated with a Machine\_mounting are the following:

- dimensions;
- location;
- machine\_mounting\_type;
- mount\_with;
- material\_type.

#### **4.2.173.1 dimensions**

The dimensions specifies the size of the mounting plate, board, or bars that secure the core box to the core machine. See 4.3.264 for the application assertion.

#### **4.2.173.2 location**

The location specifies the placement of the mounting plate, board, or bars to secure the core box to the core machine. See 4.3.265 for the application assertion.

NOTE sometimes the bars are cast on the core box as an integral part of the core box.

#### **4.2.173.3 machine\_mounting\_type**

The machine\_mounting\_type specifies specifies the method of mounting a core box on a core machine or a die on a machine for securing die tooling.

NOTE the core box may be mounted for a horizontal invest, or it may be mounted for a vertical invest or it may mounted for a top/bottom invest.

#### **4.2.173.4 material\_type**

The material\_type specifies the kind of material used for securing the core box on the core machine.

EXAMPLE materials often used are Aluminum, Steel, Cast Iron, Plywood, or Hardwood.

### **4.2.173.5 mount\_with**

The `mount_with` specifies how a tool is secured to a machine. See 4.3.xxxx for the application assertion.

NOTE See 4.2.173.5.1 to 4.2.173.5.3 for the definition of each allowable value for `mount_types`.

**4.2.173.5.1 clamps:** devices to hold firmly or immobilize a tool on a machine

**4.2.173.5.2 holes:** penetrations of some size going through a flange, bar, lug or other solid attached to a tool for the purpose of securing a tool to a machine.

**4.2.173.5.3 slots:** slits or long openings in to which something fits or in which something works for the purpose of securing a tool to a machine.

### **4.2.174 Machining\_activity**

In `Machining_activity` is a type of `Non_casting_operations` (see 4.2.202) that define any activity that interacts with the part being machined. The data associated with a `Machining_activity` are the following:

— `iso_10303_240`.

#### **4.2.174.1 iso\_10303\_240**

The `iso_10303_240` specifies the process plan data file format standard for macro process planning. See 4.3.266 for the application assertion.

### **4.2.175 Machining\_allowance**

A `Machining_allowance` is a type of `Customer_casting_requirements` (see 4.2.62) that is additional material dimensions on a cast part to allow for machining. This `Machining_allowance` is in addition to draft and shrinkage.

EXAMPLE If the part is to be machined to a 12 inch diameter, the pattern dimensions may have to be 12 3/16. The 3/16 inch allowance for shrinkage plus 1/16 inch for metal to come off. So we need an actual diameter on the small end of the pattern of 12 3/8 inches.

The data associated with a `Machining_allowance` are the following:

— `allowance_value`;

— `associated_feature`.

#### **4.2.175.1 allowance\_value**

The `allowance_value` specifies the numerical quantity of material thickness that characterizes the `Required_machining_allowance`. See 4.3.268 for the application assertion.

### 4.2.175.2 associated\_feature

The associated\_feature specifies the manufactured feature that has a machining\_allowance applied. See 4.3.267 for the application assertion.

### 4.2.176 Manufacturing\_design\_feature

A Manufacturing\_design\_feature is a type of Feature (see 4.2.124) or a Casting\_design\_feature (see 4.2.22) that identifies the types of features necessary to design, manufacture and machine a tool or a cast part. Each Manufacturing\_design\_feature is a Manufacturing\_feature (see 4.2.178)

### 4.2.177 Machining\_feature

A Machining\_feature is a type of Manufacturing\_feature (see 4.2.178) that identifies a volume of material that shall be removed to obtain the final part geometry from the cast part. Machining\_features requires both direction and location in placing them on a part. Each Machining\_feature may be a Compound\_feature (see 4.2.40).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.262.

NOTE A complete set of machining features are found in ISO 10303-224:2006 (ISO 10303 522:2006) and are the basis for application objects Design\_part and Cast\_part.

The data associated with a Machining\_feature are the following:

- placement;
- usage\_name.

#### 4.2.177.1 placement

The placement specifies the position and orientation of a Machining\_feature relative to the base shape for a part. See 4.3.269 for the application assertion.

#### 4.2.177.2 usage\_name

The usage\_name specifies a user defined name that is additional information about the use of a feature. The usage\_name need not be specified for a particular Machining\_feature.

## **4.2.178 Manufacturing\_feature**

A `Manufacturing_feature` is a type of `Manufacturing_design_feature` (see 4.2.176) that identifies the types of features necessary to manufacture and machine a part. Each `Manufacturing_feature` is either a `Machining_feature` (see 4.2.177) or a `Transition_feature` (see 4.2.335)

NOTE The application object definition is derived from ISO 10303-224:2006, Clause 4.2.123.

The data associated with a `Manufacturing_feature` are the following:

— `required_geometry`.

### **4.2.178.1 required\_geometry**

The `required_geometry` specifies the geometry necessary to define `Manufacturing_process_feature` shall be `Advanced_B_rep`. A `required_geometry` need not be specified for a particular `Manufacturing_process_feature`.

## **4.2.179 Manufacturing\_feature\_group**

The `Manufacturing_feature_group` specifies the collection of manufacturing features with a usage identification. A `Manufacturing_feature_group` shall allow for the collection of collections.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.130.

EXAMPLE A `Manufacturing_feature_group` may be used to group together all of the hold down holes for a part.

The data associated with a `Manufacturing_feature_group` are the following:

- `feature_groups`;
- `group_description`;
- `group_name`.

### **4.2.179.1 feature\_groups**

The `feature_group` specifies the list of `Manufacturing_feature` (see 4.2.178) or the `Manufacturing_feature_group` (see 4.2.179) to be grouped. See 4.3.270 and 4.3.271 for the application assertion.

### **4.2.179.2 group\_description**

The `group_description` specifies additional text information about the `Manufacturing_feature_group`.

### 4.2.179.3 group\_name

The group\_name specifies identification for the Manufacturing\_feature\_group.

## 4.2.180 Manufacturing\_process

The Manufacturing\_process specifies a sequential list of manufacturing activities to be performed by a specific machine with a specific machine\_setup necessary to manufacture a part. A Manufacturing\_process is either an Equipment\_process (see 4.2.111), or a Non\_equipment\_process (see 4.2.203).

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.77.

The data associated with a Manufacturing\_process are the following:

- assigned\_operation;
- description;
- name;
- setup;

### 4.2.180.1 assigned\_operation

The assigned\_feature specifies Casting\_activity (see 4.2.21) that are components of the shape of the Part. The assigned\_feature need not be specified for a particular Manufacturing\_process. See 4.3.273 for the application assertion.

### 4.2.180.2 description

The description specifies the word, or group of words, that explain the single setup. A description need not be specified for a particular Manufacturing\_process.

### 4.2.180.3 name

The name specifies the word, or group of words, that make up the unique designation of a single setup.

### 4.2.180.4 setup

The setup specifies the Setup\_activity (see 4.2.280) that defines a specific machine setup for a machining process. See 4.3.272 for the application assertion.

## 4.2.181 Master\_pattern

A `Master_pattern` is a type of `Sand_casting_tooling` (see 4.2.277) that is a scaled replica used for making a casting that will become the production pattern needed to produce the cast part. The scale factor will be determined by the cast part material together with the material and the process used to produce the `Production_pattern`.

EXAMPLE A master pattern need to produce a production pattern in an Aluminum match-plate where the cast part is of cast brass. The master pattern will be scaled by a factor of 1.026042" to produce the cast part to specified design dimensions where dimensions are in inches.

The data associated with a `Master_pattern` are the following:

— `machining_allowance_on_master_pattern`;

### 4.2.181.1 machining\_allowance\_on\_master\_pattern

The `machining_allowance_on_master_pattern` specifies the stock to be added to the tool to permit machining. There may be more than one `machining_allowance_on_master_pattern` specified for a `Master_pattern`. See 4.3.274 for the application assertion.

## 4.2.182 Master\_sample

In `Master_sample` is a type of `Cast_part`(see 4.2.19) that is an inspected part that has been produced by a casting process and is identified by the supplier as the master sample.

NOTE When a Production Part Approval Process (PPAP) is specified by the customer a master sample is required to support the Part Submission Warrant (PSW) approval for the part.

The data associated with an `Master_sample` are the following:

— `belongs_to_heat`;

— `belongs_to_lot`;

— `process_plan_used`;

— `production_date`;

— `product_specification_used`.

### 4.2.182.1 belongs\_to\_heat

The `belongs_to_heat` specifies the `Heat` (see 4.2.143) to which an `Master_sample` belongs. The `belongs_to_heat` need not be specified for a particular `Master_sample`. See 4.3.275 for the application assertion.

#### **4.2.182.2 belongs\_to\_lot**

The `belongs_to_lot` specifies the Lot (see 4.2.172) to which an `Master_sample` belongs. The `belongs_to_lot` need not be specified for a particular `Master_sample`. See 4.3.276 for the application assertion.

#### **4.2.182.3 process\_plan\_used**

The `process_plan_used` specifies the operations by which the `Master_sample` was manufactured. See 4.3.280 for the application assertion.

#### **4.2.182.4 production\_date**

The `production_date` specifies the date and time when the `Master_sample` was created. See 4.3.277 for the application assertion.

#### **4.2.182.5 product\_specification\_used**

The `product_specification_used` specifies the design that the `Master_sample` was built to satisfy. See 4.3.279 or 4.3.278 for the application assertion.

### **4.2.183 Material**

A `Material` is the identification of the raw stock that is the source object the process plan is acting upon.

NOTE This application object is harmonized with the same from ISO 10303-224:2006, Clause 4.2.132.

The data associated with a `Material` are the following:

- composition;
- `material_characteristics`;
- `material_description`;
- `material_id`;
- `material_specification`;
- structure.

### 4.2.183.1 composition

The composition specifies the actual material content of a Material. The composition need not be specified for a particular Material. There may be more than one composition for a Material. See 4.3.281 for the application assertion.

### 4.2.183.2 material\_description

The material\_description specifies a user defined explanation of the material required for the part.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for core\_box\_material type

**4.2.183.2.1 aluminum:** an non-ferrous material used to produce production dies, patterns or core box tooling.

NOTE The pattern or core box could be machined from a block of aluminums or cast to near shape.

**4.2.183.2.2 brass:** a non-ferrous material used in the production of tooling for cast parts or as inserts in the cast part.

**4.2.183.2.3 ceramic:** a refractory material use in the production of some investment type casting or as a slurry coating for cores.

**4.2.183.2.4 epoxy:** a plastic material used to produce master tooling, some production pattern and core box tooling.

NOTE the epoxy plastic could be with a surface coat with a fiber reinforced backing to the surface coat.

**4.2.183.2.5 expandable\_polystyrene:** a polymerized styrene material in bead form that when heated to a specified temperature expands. A material used for lost foam casting processes.

**4.2.183.2.6 hard\_wood:** a high density wood material used to produce master tooling, some production pattern or core box tooling.

**4.2.183.2.7 iron:** The pattern or core box could be machined from a block of solid cast iron or cast to near shape.

**4.2.183.2.8 other:** any other material not identified above used to produce a die, pattern or core box tooling.

**4.2.183.2.9 plaster:** the material used to produce patterns or core box tooling.

**4.2.183.2.10 plywood:** the material used to produce the pattern or core box tooling.

**4.2.183.2.11 polymethylmethacrylate:** a material that is the most widely used alternative to polystyrene in the lost foam process.



**4.2.183.2.12 shell:** a material used with investment casing processes. Or form of moulding process with sand castings.

**4.2.183.2.13 soft\_wood:** the low density wood material used to produce the pattern or core box tooling.

**4.2.183.2.14 steel:** The pattern or core box could be machined from a block of steel or cast to near shape.

**4.2.183.2.15 urethane:** the plastic material used to produce the pattern or core box tooling.

**4.2.183.2.16 wood\_metal\_reinforced:** the material used to produce the pattern or core box tooling.

NOTE the tooling may be constructed primarily of hard or soft wood but faced or braced with metal to add durability to the tooling.

### **4.2.183.3 material\_characteristics**

The `material_characteristics` specifies the properties which define the Material. The `material_characteristics` need not be specified for a particular Material. There may be more than one `material_characteristics` for a Material. See 4.3.282 for the application assertion.

### **4.2.183.4 material\_id**

The `material_id` specifies a word or group of words that make up the unique designation of the material.

### **4.2.183.5 material\_specification**

The `material_specification` specifies the documentation which contains additional information about Material. The `material_specification` need not be specified for a particular Material. There may be more than one `material_specification` for a Material. See 4.3.284 for the application assertion.

### **4.2.183.6 structure**

The `structure` specifies the composition of the Material in terms of smaller physical units. The `structure` need not be specified for a particular Material. See 4.3.283 for the application assertion.

## **4.2.184 Material\_composition**

A `Material_composition` is the description of the chemical elements used to define a specific Material (see 4.2.183). The data associated with a `Material_composition` are the following:

- `class`;
- `composition_basis`;
- `description`;

ISO 10303-223:2008 (E)

- determination\_method;
- elements;
- name.

#### **4.2.184.1 class**

The class specifies the name or identi\_er for the kind of relationship between a constituent and a product.

EXAMPLE Possible values for class include mixture, chemically bonded, and alloyed.

#### **4.2.184.2 composition\_basis**

The composition\_basis specifies the basis on which a product is decomposed into its constituents.

EXAMPLE Expected values for composition basis include volume, weight, moles and atoms.

#### **4.2.184.3 elements**

The elements specifies the list of chemical elements that define the composition of a material. See 4.3.285 for the application assertion.

#### **4.2.184.4 description**

The description specifies a user defined information about the material composition.

#### **4.2.184.5 determination\_method**

The determination\_method specifies a description of the procedure by which constituent amount is determined.

#### **4.2.184.6 name**

The name specifies a word or group of words by which an material composition is commonly referred.

### 4.2.185 Material\_condition\_modifier

A `Material_condition_modifier` is a description of a condition for a part surface, it is an indication whether the maximum material principle is to be applied respectively to the toleranced feature, the datum feature, or both. Tolerances of position are determined by a material condition of a mating surface.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.133.

The data associated with a `Material_condition_modifier` are the following:

- `material_type`.

#### 4.2.185.1 material\_type

The `material_type` specifies the type of condition allowed for the `Material_condition_modifier`.

EXAMPLE maximum material condition (MMC), least material condition (LMC) and regardless of feature size (RFS).

### 4.2.186 Material\_property

A `Material_property` is the information that describes the properties of the material used to produce a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.134.

The data associated with a `Material_property` are the following:

- `alloy_grade`;
- `material_hardness`;
- `melt_temperature`;
- `property_characteristics`.

#### 4.2.186.1 alloy\_grade

The `alloy_grade` specifies the words or the group of words that describes the classification of a substance that has calibrated metallic properties and that is composed of two or more chemical elements.

EXAMPLE Class 30 iron, Meehanite grade A, AL-365 are examples of alloy grades.

#### 4.2.186.2 material\_hardness

The `material_hardness` specifies additional information to define hardness properties. The `material_hardness` need not be specified for a particular `Material_property`. There may be more than one `material_hardness` for a `Material_property`. See 4.3.286 for the application assertion.

### **4.2.186.3 melt\_temperature**

The `melt_temperature` specifies the temperature when a material goes from a solid state to liquid or fluid state. See 4.3.287 for the application assertion.

### **4.2.186.4 property\_characteristics**

The `property_characteristic` specifies the parameter for the description of `Material_property`. The `property_characteristics` need not be specified for a particular `Material_property`. There may be more than one `property_characteristics` for a `Material_property`. See 4.3.288 for the application assertion.

### **4.2.187 Material\_requisition**

A `Material_requisition` is a type of `Requisition` (see 4.2.265) which describes an order for purchase of raw materials by the manufacturing organization.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.135.

### **4.2.188 Material\_structure**

A `Material_structure` is the representation of the physical characteristics of a `Material` (see 4.2.183). The data associated with a `Material_structure` are the following:

- `element`;
- `illustration`;
- `reference_to`.

#### **4.2.188.1 element**

The `element` specifies the chemical element that is contained in a `Material_structure` (see 4.2.188). There may be more than one `element` for a `Material_structure_element`. See 4.3.290 for the application assertion.

#### **4.2.188.2 illustration**

The `illustration` specifies the an image that illustrates the `Material_structure`. The `illustration` need not be specified for a particular `Material_structure`. See 4.3.289 for the application assertion.

#### **4.2.188.3 reference\_to**

The `reference_to` specifies the specification or standard specifying the `Material_structure`. The `reference_to` need not be specified for a particular `Material_structure`. See 4.3.291 for the application assertion.

## 4.2.189 Material\_structure\_element

A Material\_structure\_element is a physical constituent of a Material (see 4.2.183). The data associated with a Material\_structure\_element are the following:

- amount;
- characteristic\_geometry;
- made\_of;
- type\_of\_element.

### 4.2.189.1 amount

The amount specifies the quantitative measure of the physical constituent. See 4.3.294 for the application assertion.

### 4.2.189.2 characteristic\_geometry

The characteristic\_geometry specifies the size, shape and orientation of the physical constituent. The characteristic geometry need not be specified for a particular Material\_structure\_element. See 4.3.292 for the application assertion.

### 4.2.189.3 made\_of

The made\_of specifies the substance comprising the physical constituent. See 4.3.293 for the application assertion.

### 4.2.189.4 type\_of\_element

The type\_of\_element specifies the kind of Material\_structure\_element.

## 4.2.190 Material\_usage

A Material\_usage is the kind and amount of material used in a Activity (see 4.2.1). The data associated with a Material\_usage are the following:

- amount;
- actual\_amount.

### 4.2.190.1 actual\_material

The actual\_material specifies the material to be employed. See 4.3.296 for the application assertion.

## **4.2.190.2 amount**

The amount specifies the quantitative measure of material to be employed. See 4.3.295 for the application assertion.

## **4.2.191 Material\_usage\_record**

A `Material_usage_record` is a type of `Activity_result_record` (see 4.2.3) that is a log of material used during a `Activity` (see 4.2.1).

NOTE A `Material_usage_record` may be used for later inspection and analysis to determine the relationship between manufacturing variables and product quality.

The data associated with a `Material_usage_record` are the following:

- `actual_substance_usage_amount`;
- `recorded_usage`.

### **4.2.191.1 actual\_substance\_usage\_amount**

The `actual_substance_usage_amount` specifies the actual amount of material used during performance of a `Activity` (see 4.2.1). See 4.3.297 for the application assertion.

### **4.2.191.2 recorded\_usage**

The `recorded_usage` specifies the `Material_usage` (see 4.2.190) that the `Material_usage_record` is being applied. See 4.3.298 for the application assertion.

## **4.2.192 Mechanical\_property\_requirements**

The `Mechanical_property_requirements` are those property requirements of a material that reveal the elastic and inelastic properties when force is applied.

NOTE These requirements should not be used or confused with “physical property requirements”.

The data associated with a `Mechanical_property_requirements` are the following:

- `chemical_analysis_certification`.
- `material_hardness`;
- `mechanical_properties_certification`;

- requirement\_specification;
- tensile\_strength;
- yeild\_strength.

#### **4.2.192.1 chemical\_analysis\_certification**

The chemical\_analysis\_certification specifies a Boolean value that indicates if the chemical composition for the casting is met and certified. A value of true specifies that the chemical\_analysis is met and certified. A value of false specifies that the chemical\_analysis is not met and certified.

#### **4.2.192.2 material\_hardness**

The material\_hardness specifies the range of values or the maximum hardness value of the material the customer will accept for the cast part. See 4.3.299 for the application assertion.

NOTE the value is determined by the Rockwell, Brinell, or Vickers standards.

#### **4.2.192.3 mechanical\_properties\_certification**

The mechanical\_properties\_certification specifies a Boolean value that indicates if the mechanical properties for the casting are met and certified. A value of true specifies that the mechanical properties are met and certified. A value of false specifies that the mechanical properties not are met and certified.

#### **4.2.192.4 requirement\_specification**

The requirement\_specification specifies one or more document or standard where the specification requirements for the Mechanical\_property\_requirements are defined. See 4.3.301 for the application assertion.

#### **4.2.192.5 tensile\_strength**

The tensile\_strength specifies the maximum stress in uniaxial tension testing that a material will withstand prior to fracture. The stress load is given as a numeric value with units for the material tested. The tensile\_strength need not be specified for a particular Heat\_treat\_requirement for the cast part. See 4.3.300 for the application assertion.

#### **4.2.192.6 yield\_strength**

The yield\_strength specifies the stress at which a material exhibits a specified limiting permanent strain or deformation. See 4.3.300 for the application assertion.

NOTE yield strength is the point where a material bends or deforms and the material will not return to its original shape.

### **4.2.193 Mechanical\_stress**

The `Mechanical_stress` is the type of design load with the maximum load value an in-service application will apply to the cast part. The data associated with a `Mechanical_stress` are the following:

- `maximum_psi_design_stress`;
- `percent_safety_factor_required`;
- `type_of_stress`.

#### **4.2.193.1 maximum\_psi\_design\_stress**

The `maximum_psi_design_stress` specifies the type of maximum design loads for mechanical properties for which the cast part is designed. The `maximum_psi_design_stress` need not be specified for a particular `Mechanical_stress` for the cast part. See 4.3.303 for the application assertion.

#### **4.2.193.2 percent\_safety\_factor\_required**

The `percent_safety_factor_required` specifies the factor of safety for the application loads for which the cast part is designed. The `percent_safety_factor_required` need not be specified for a particular `Mechanical_stress` for the cast part. See 4.3.303 for the application assertion.

#### **4.2.193.3 type\_of\_stress**

The `type_of_stress` specifies the type of design mechanical properties with numerical data for which the cast part is designed. The `type_of_stress` need not be specified for a particular `Mechanical_stress` for the cast part. See 4.3.302 for the application assertion.

#### **4.2.194 Melting\_equipment**

The `Melting_equipment` is a type of `Equipment` (see 4.2.110) that produces enough heat to change a solid material to a liquefied state.

#### **4.2.195 Meshing\_condition**

A `Meshing_condition` is a description of the fineness of mesh needed in a region. The data associated with a `Meshing_condition` are the following:

- `area_to_be_meshed`;
- `density_of_elements`;



- density\_of\_nodes;
- number\_of\_elements;
- number\_of\_nodes.

#### **4.2.195.1 area\_to\_be\_meshed**

The areas to be meshed specifies the region to which the Meshing\_condition applies. The areas to be meshed need not be specified for a particular Meshing\_condition. See 4.3.305 for the application assertion.

NOTE A Meshing\_condition may apply to an entire Part\_version (see 4.2.217), or a portion of an Item.

#### **4.2.195.2 density\_of\_elements**

The density of elements specifies the number of elements per unit volume for the region. The density of elements need not be specified for a particular Meshing\_condition. See 4.3.304 for the application assertion.

#### **4.2.195.3 density\_of\_nodes**

The density of nodes specifies the number of nodes per unit volume for the region. The density of nodes need not be specified for a particular Meshing\_condition. See 4.3.304 for the application assertion.

#### **4.2.195.4 number\_of\_elements**

The number\_of\_elements specifies the quantity of elements for region.

#### **4.2.195.5 number\_of\_nodes**

The number\_of\_nodes specifies the quantity of nodes for region.

### **4.2.196 Metal\_caster\_simulation**

The Metal\_caster\_simulation is a type of simulation\_process (see 4.2.295) that request from a foundry to use a simulation software tool to produce a simulation of a casting process for a proposed cast part.

ISO 10303-223:2008 (E)

The data associated with an `Metal_caster_simulation` are the following:

- `available_tooling`;
- `defect_results`;
- `proposed_gating`;
- `suggestion_for_changes`;

#### **4.2.196.1 available\_tooling**

The `available_tooling` specifies what pattern equipment and that that is available. See 4.3.309 for the application assertion.

#### **4.2.196.2 defect\_results**

The `defect_results` specifies a list of one or more defects identified by the simulation software run and the data analysis from the simulation run. See 4.3.307 for the application assertion.

#### **4.2.196.3 proposed\_gating**

The `proposed_gating` specifies the existing gating components with parameters or the proposed gating components with parameters. See 4.3.308 for the application assertion.

#### **4.2.196.4 suggestion\_for\_changes**

The `suggestion_for_changes` specifies changes that could be made to the tooling or the part design as identified by the simulation software run and the data analysis from the simulation run. See 4.3.306 for the application assertion.

#### **4.2.197 Mismatch\_tolerance**

A `Mismatch_tolerance` is an imperfection on a cast part caused by inaccurate alignment or displacement or dimensional differences between the mould or die components, producing a step on the surface along the parting line.

NOTE This application object is harmonized with the dimensional mismatch object from ISO 8062-1:2007, Clause 4.2.3.

The data associated with a `Mismatch_tolerance` are the following:

- `maximum_amount`;
- `with_respect_to`.

#### 4.2.197.1 maximum\_amount

The maximum\_amount specifies the largest allowable amount of mismatch. See 4.3.310 for the application assertion.

#### 4.2.197.2 with\_respect\_to

The with\_respect\_to specifies the Parting\_surface (see 4.2.218) with which the Mismatch\_tolerance is associated. The with\_respect\_to need not be specified for a particular Mismatch\_tolerance. See 4.3.311 for the application assertion.

NOTE In castings which have more than one parting surface, it may be desirable to specify a different amount of mismatch along each surface.

#### 4.2.198 Model\_element

A Model\_element is a portion of a model representation.

NOTE This application object is harmonized with the same from ISO 10303-240:2005, Clause 4.2.85.

The data associated with a Model\_element are the following:

- element;
- representation\_type.

##### 4.2.198.1 element

The element specifies the portion of the Geometric\_model that defines the Model\_element. See 4.3.312 for the application assertion.

##### 4.2.198.2 representation\_type

The representation\_type specifies the geometric representations that define a shape. See 4.3.313 for the application assertion.

#### 4.2.199 Mould\_box

A Mould\_box is a type of Sand\_casting\_design\_feature (see 4.2.276) that is a container that is designed to mould a cast part that requires no metal jacket around the mould when the mould box is removed from the mould. The data associated with a Mould\_box are the following:

- box\_size;
- description.

ISO 10303-223:2008 (E)

#### **4.2.199.1 box\_size**

The `box_size` specifies the dimensions of the container used to retain the moulding material to form the mould for the cast part. See 4.3.314 for the application assertion.

#### **4.2.199.2 description**

The description specifies the style of the mould box. See 4.3.315 for the application assertion.

#### **4.2.200 Mould\_locks**

The `Mould_locks` is a type of `Sand_casting_design_feature` (see 4.2.276) that are forms located, inside the flask line, on the pattern parting board or plate that makes an impression in the mould sand that will align the cope and drag moulds, when the moulds are closed, to insure a match of the cast part at the parting line.

EXAMPLES - `Mould_locks` are often referred to as shift locks, chain locks, etc.

NOTE When shift locks buttons are specified the point location is required for each shift lock button. When chain or wedge locks are specified the maximum rectangular dimensional size is required.

The data associated with a `Mould_locks` are the following:

- `description`;
- `lock_dimensions`.

##### **4.2.200.1 description**

The description specifies the style of the mould lock. See 4.3.317 for the application assertion.

EXAMPLE styles are shift lock button, chain lock, print and core are examples of `mould_locks`.

##### **4.2.200.2 lock\_dimensions**

The `lock_dimensions` specifies the measurement values for the mould lock. See 4.3.316 for the application assertion.

#### **4.2.201 Moulding\_equipment**

A `Moulding_equipment` is a type of `Equipment` (see 4.2.110) that is the machinery used to produce sand moulds from production patterns.

## 4.2.202 Non\_casting\_operations

A Non\_casting\_operations is a type of Casting\_activity (see 4.2.21) that identifies activities to be performed on the cast part that are outside of metal caster activities. Each Non\_casting\_operations is either a Assembly (see 4.2.13), Heat\_treat( see 4.2.144), Fabrication (see 4.2.121), Impregnation\_activity (see 4.2.148), Machining\_activity (see 4.2.174), Production\_welding (see 4.2.249), or Surface\_finish (see 4.2.316).

The data associated with a Non\_casting\_operations are the following:

- customer\_requirements;
- operation\_description;
- part\_geometry\_with\_machine\_allowance;
- part\_geometry\_with\_no\_machine\_allowance.

### 4.2.202.1 customer\_requirements

The customer\_requirements specifies specific machining and finishing procedures to be completed on a casting prior to delivery of the cast part. The customer\_requirements need not be specified for a particular Non\_casting\_operations. See 4.3.321 for the application assertion.

### 4.2.202.2 operation\_description

The operation\_description specifies an action, process, or activity to be completed before delivery of the cast part. The operation\_description need not be specified for a particular Non\_casting\_operations. See 4.3.320 for the application assertion.

### 4.2.202.3 part\_geometry\_with\_machine\_allowance

The part\_geometry\_with\_machine\_allowance specifies the surfaces that have additional material added for machined surfaces. The part\_geometry\_with\_machine\_allowance need not be specified for particular Non\_casting\_operations. See 4.3.318 for the application assertion.

### 4.2.202.4 part\_geometry\_with\_no\_machine\_allowance

The part\_geometry\_with\_no\_machine\_allowance specifies specifies the surfaces that have no additional material added to a surface. The part\_geometry\_with\_no\_machine\_allowance need not be specified for particular Non\_casting\_operations. See 4.3.319 for the application assertion.

### 4.2.203 Non\_equipment\_process

A Non\_equipment\_process is a type of Manufacturing\_process (see 4.2.180) that specifies the type of equipment setup, and equipment activity, required to perform a non-automated machining process.

### 4.2.204 Non\_permanent\_moulding\_process

A Non\_permanent\_moulding\_process is a type of Casting\_activity (see 4.2.21) that is a process where the mould is destroyed when the metal casting has been removed from the mould. The data associated with a Non\_permanent\_moulding\_process are the following:

- flask\_type;
- moulding\_process;
- number\_of\_impressions;
- pattern\_tooling;
- pattern\_form.

#### 4.2.204.1 flask\_type

The flask\_type specifies the flask used to mould the pattern tooling to produce the cast part. Each flask\_type may be either Flask or Flaskless. See 4.3.322, 4.3.323 for the application assertion.

#### 4.2.204.2 number\_of\_impressions

The number\_of\_impressions specifies the total number of patterns attached to a pattern mould board or attached to a mould cavity tree.

#### 4.2.204.3 moulding\_process

The moulding\_process specifies the system operations used to produce the nonpermanent mould for the cast part.

NOTE See 4.2.204.3.1 to 4.2.204.3.5 for the definition of each allowable value for moulding\_process types.

**4.2.204.3.1 ceramic:** a moulding process where a refractory slurry is applied to a pattern for a specified shell thickness and heated to specified temperature to cure the shell at which time the pattern will melt or evaporate when heat is applied.

**4.2.204.3.2 green\_sand:** a moulding process where the mould sand materials are mixed and held together by clay, water, and other additives.

**4.2.204.3.3 lost\_foam:** A casting process in which a foam pattern is replaced by molten metal in a flask filled with loose sand to form a casting.

**4.2.204.3.4 no\_bake:** where moulds or cores are produced with a resin-bonded air-setting sand. Also known as the air set process because moulds are left to harden under atmospheric conditions.

**4.2.204.3.5 shell:** A process for forming a mould from resin-bonded sand mixtures brought in contact with preheated (300-500F metal patterns, resulting in a firm shell with a cavity corresponding to the outline of the pattern.

#### **4.2.204.4 pattern\_tooling**

The `pattern_tooling` specifies the type of impression and shape contour used to form the mould cavity for the cast part. See 4.3.324, 4.3.326 and 4.3.325 for the application assertion.

#### **4.2.204.5 pattern\_form**

The `pattern_form` specifies the type of shape and configuration used to form the mould cavity for the cast part.

NOTE See 4.2.204.5.1 to 4.2.204.5.10 for the definition of each allowable value for `pattern_form` type.

**4.2.204.5.1 loose\_pattern\_with\_follow\_on\_board:** a solid pattern that can not be mounted on a pattern board, but has a board that is hallowed out as pattern bed such that a natural parting line is formed so the drag and cope can be easily moulded.

**4.2.204.5.2 loose\_solid\_pattern:** a solid pattern that can not be mounted on a pattern board.

**4.2.204.5.3 loose\_split\_pattern:** a split pattern that can be mounted on a pattern board or plate.

**4.2.204.5.4 mounted\_cope\_and\_drag\_boards:** the cope half of the pattern is secured on a separate board to form the cope mould and the drag half secured on a separate board to form the drag mould. Each pattern half shall be matched to the pin centers of the board for the required flask.

NOTE boards could of wood, metal plates, or plastic

**4.2.204.5.5 mounted\_cope\_and\_drag\_flaskless:** the cope half of the pattern is secured on a separate board that is attached to a wooden box or metal tub to form the cope mould and the drag half of the pattern is secured on a separate board that is attached to a wooden box or metal tub to form the drag mould. Each pattern half shall be matched through some register mechanism as there are no flask matching pins with this moulding process.

NOTE boards could wooden, metal plates, or plastic

ISO 10303-223:2008 (E)

**4.2.204.5.6 mounted\_cope\_and\_drag\_inserts:** the cope half of the pattern is secured on a separate board of predetermined size, the drag half secured on a separate board of predetermined size. Each insert board is attached to an automatic moulding machine.

NOTE boards could of wood, metal plates, or plastic

**4.2.204.5.7 mounted\_match\_plate:** cope and drag pattern(s) are secured, back to back split along the parting line, to a metal plate.

**4.2.204.5.8 mounted\_snapboard:** cope and drag pattern(s) are secured, back to back split along the parting line, to a plywood board.

NOTE This pattern type will be for the squeezer or snap line moulding process.

**4.2.204.5.9 shell pattern:** cope and drag metal pattern(s) are secured to a metal plate with a predetermined register system for matching the cope and drag moulds for the molten pouring process.

NOTE metal pattern are of cast iron, steel, and sometimes aluminum.

**4.2.204.5.10 sweep\_pattern:** a pattern that is made by the use of a template formed to a required shape or profile that is moved along a specified path to sweep a form for the pattern shape.

## 4.2.205 Numeric\_parameter

A Numeric\_parameter is a type of Property\_parameter (see 4.2.256) that a numeric value with units of the property being defined. A Numeric\_parameter is either a Numeric\_range (see 4.2.207) or a Numeric\_parameter\_with\_tolerance (see 4.2.206).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.143.

The data associated with a Numeric\_parameter are the following:

- parameter\_description;
- parameter\_note;
- parameter\_units;
- parameter\_value
- significant\_digits.

### 4.2.205.1 parameter\_description

The parameter\_description specifies textural information to add user description to the use of a specific Numeric\_parameter.



#### 4.2.205.2 parameter\_note

The parameter\_note specifies textural information to supply a user defined note to the use of a specific Numeric\_parameter.

#### 4.2.205.3 parameter\_units

The parameter\_units specifies the quantity of measure in which the value is given.

EXAMPLE watt, meters, degrees, etc.

#### 4.2.205.4 parameter\_value

The parameter\_value specifies the numeric amount associated with the units of a specific characteristic of interest.

#### 4.2.205.5 significant\_digits

The significant\_digits specifies the number of decimal places indicating the accuracy of value. Significant\_digits need not be specified for a particular Numeric\_parameter.

### 4.2.206 Numeric\_parameter\_with\_tolerance

A Numeric\_parameter\_with\_tolerance is a type of Numeric\_parameter (see 4.2.205) with an implied tolerance value.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.144.

NOTE 2 A thread has a implicit definition for the minor\_diameter attribute. This attribute has no explicit geometry definition, so the dimensional tolerance of this attribute is represented with Numeric\_parameter\_with\_tolerance.

The data associated with a Numeric\_parameter\_with\_tolerance are the following:

— implicit\_tolerance.

#### 4.2.206.1 implicit\_tolerance

The implicit\_tolerance specifies the type of tolerance to apply to a numeric parameter value. See 4.3.327 and 4.3.327, 4.3.328, 4.3.329 and 4.3.330 for the application assertion.

### 4.2.207 Numeric\_range

A Numeric\_range is a type of Numeric\_parameter (see 4.2.205) that is a range of values and a desired nominal value specified for a physical characteristic or property. The data associated with a Numeric\_range are the following:

— value\_range.

© ISO 2008 — All rights reserved

185

#### **4.2.207.1 value\_range**

The `value_range` specifies the upper and lower range for the numeric value. See 4.3.331 for the application assertion.

#### **4.2.208 Object\_element\_shape\_representation**

An `Object_element_shape_representation` is a grouping of representations to define a shape.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.91.

The data associated with an `Object_element_shape_representation` are the following:

- `representation_type`;
- `shape_definition`.

#### **4.2.208.1 representation\_type**

The `representation_type` specifies the geometric representations that define a shape. See 4.3.332 for the application assertion.

#### **4.2.208.2 shape\_definition**

The `shape_definition` specifies the shape that is the representation. See 4.3.333 for the application assertion.

#### **4.2.209 Ordered\_part**

An `Ordered_part` specifies the number of machined parts, of a particular order number, a `Customer_order` is requesting.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.148.

The data associated with an `Ordered_part` are the following:

- `quantity_required`;
- `quantity_unit_of_measure`.

#### **4.2.209.1 quantity\_required**

The `quantity_required` specifies the number of items to be manufactured.

### 4.2.209.2 quantity\_unit\_of\_measure

The quantity\_unit\_of\_measure specifies the units in terms of which the part is expressed.

EXAMPLE Examples of units of measure are: each, meter, watt, etc.

### 4.2.210 Organization

An Organization is a functional group that is involved in the production of a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.149.

The data associated with an Organization are the following:

- address;
- division;
- organization\_id;
- organization\_name.

#### 4.2.210.1 address

The address specifies a location at which an organization exists and may be contacted. This location may be physical or electronic. See 4.3.334 for the application assertion.

#### 4.2.210.2 division

The division specifies a major unit of people authorized to act with specific responsibility within an organization or corporation.

#### 4.2.210.3 organization\_id

The organization\_id specifies the unique identification of an organization.

#### 4.2.210.4 organization\_name

The organization\_name specifies a word or group of words by which an organization is commonly referred. This name need not be unique within an organization.

### **4.2.211 Orientation**

An Orientation is the direction and location of the basic shape of a part, feature on the part, or of the component of a feature which are Profile objects and Path objects.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.150.

The data associated with an Orientation are the following:

- axis;
- location.

#### **4.2.211.1 axis**

The axis specifies a line in 3D space about which the part or portions of the part are arranged.

#### **4.2.211.2 location**

The location specifies a point in 3D space used to position the part or portions of the part.

### **4.2.212 Other\_inspection**

The Other\_inspection is a type of Inspection\_or\_test\_result (see 4.2.155) that is the outcome of an examination or measurement and comparing the resultant to the design requirement.

The data associated with an Other\_inspection are the following:

- requirement;
- test\_result.

#### **4.2.212.1 inspection\_result**

The inspection\_result specifies one or more of the outcomes of a specific inspection against the customer requirement for that inspection.

#### **4.2.212.2 requirement**

The requirement specifies the Inspection\_or\_test\_requirement (see 4.2.154) that was evaluated to produce the result. See 4.3.335 for the application assertion.

### 4.2.213 Part\_dimensioning\_standard

A `Part_dimensioning_standard` is a type of `Document_assignment` (see 4.2.94) that is the reference to a document that defines the standard used to define the dimension tolerance used.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.156.

The data associated with a `Part_dimensioning_standard` are the following:

— `applied_part`

#### 4.2.213.1 applied\_part

The `applied_part` specifies the part that uses the dimensioning standard. See 4.3.336 for the application assertion.

### 4.2.214 Part\_property

A `Part_property` is a specific characteristic about the form, fit, or function of a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.158.

The data associated with a `Part_property` are the following:

— `property_characteristic`.

#### 4.2.214.1 property\_characteristic

The `property_characteristic` specifies the parameter to describe the `Part_property`. The `property_characteristic` need not be specified for a particular `Part_property`. There may be more than one `property_characteristic` for a `Part_property`. See 4.3.337 for the application assertion.

### 4.2.215 Part\_relationship

The `Part_relationship` is relating the `Design_part` (see 4.2.76) to the `Cast_part` (see 4.2.19) where the design features are the base design features that are modified for the part to be cast.

NOTE 1 a base design feature is a feature that has no draft, shrink factor, or machining allowance. The modified base feature is related to the base design feature for machining and inspection. The `Design_part` (see 4.2.76) `Manufacturing_design_feature` (see 4.2.176) is to be related to the `Cast_part` (see 4.2.19) `Casting_design_feature` (see 4.2.22).

NOTE 2 this relationship between the features of the designed part is very critical for machined surfaces where machining allowances have been added or where feature of the designed part do not appear in the cast part but are to be machined in the casting.

EXAMPLE: a boss is a base design feature that is modified with draft, a shrink factor, and possibly machine stock if the boss is to be machined to make it a `Casting_design_feature` (see 4.2.22).

ISO 10303-223:2008 (E)

The data associated with a Part\_relationship are the following:

- as\_designed;
- to\_be\_cast;
- tooling\_equipment.

#### **4.2.215.1 as\_designed**

The as\_designed specifies a specific design feature of the Design\_part (see 4.2.76). See 4.3.338 for the application assertion.

#### **4.2.215.2 to\_be\_cast**

The to\_be\_cast specifies a specific cast feature of the Cast\_part (see 4.2.19). See 4.3.339 and 4.3.340 for the application assertion.

#### **4.2.215.3 tooling\_equipment**

The tooling\_equipment specifies the tool necessary to produce the specific cast feature required for the design feature. There may be more than one tooling\_equipment for a Part\_relationship. The tooling\_equipment need not be specified for a particular Part\_relationship. See 4.3.341 for the application assertion.

### **4.2.216 Part\_shape**

A Part\_shape is the physical form of the part that is being machined.

NOTE This application object is harmonized with the shape object from ISO 10303-224:2006, Clause 4.2.217.

The data associated with a Part\_shape are the following:

- base\_shape\_definition;
- elements;
- representation\_form.

#### **4.2.216.1 base\_shape\_definition**

The base\_shape\_definition specifies the minimum Base\_shape (see 4.2.15) of rectangular size to encapsulate a cast part. The base\_shape\_definition need not be specified for a particular Part\_shape. See 4.3.342 for the application assertion.

### 4.2.216.2 elements

The elements specifies the components of the shape of the Part. The element need not be specified for a particular Shape. See 4.3.344 for the application assertion.

### 4.2.216.3 representation\_form

The representation\_form specifies the representation to define the shape of the Part. The representation\_form need not be specified for a particular Part\_shape. There may be more than one representation\_form for a Part\_shape. See 4.3.343 for the application assertion.

### 4.2.217 Part\_version

A Part\_version is the physical item that is intended to be produced through a manufacturing process. Each Part may be one of the following: Cast\_part (see 4.2.19), Design\_part (see 4.2.76), Cast\_part\_with\_rigging (see 4.2.20), or Red\_line\_part (see 4.2.261).

NOTE This application object is harmonized with the part object from ISO 10303-224:2006, Clause 4.2.155.

The data associated with a Part\_version are the following:

- casting\_defined\_by;
- customer\_requirement;
- drawing\_info;
- manufacture\_authorization;
- manufactured\_by;
- material\_definition;
- owned\_by;
- part\_description;
- part\_id;
- part\_name;
- part\_revision\_id;
- physical\_form;

ISO 10303-223:2008 (E)

- property\_characteristics;
- quantity\_ordered;
- security\_classification;

#### **4.2.217.1 casting\_defined\_by**

The `casting_defined_by` specifies a `Process_plan` (see 4.2.236) or an `External_schema_definition` (see 4.2.xxx) that defines the corporate level process activities and procedures to manufacture the required cast part for a specific `Part_version`. See 4.3.354 and 4.3.348 for the application assertion.

#### **4.2.217.2 customer\_requirement**

The `customer_requirement` specifies an instruction or standard that shall be followed in detail for acceptance of the manufactured cast part. The `customer_requirement` need not be specified for a particular `Part_version`. There may be more than one `customer_requirement` for a `Part_version`. See 4.3.346 for the application assertion.

#### **4.2.217.3 drawing\_info**

The `drawing_info` specifies the product referenced by the `Design_reference`. There may be more than one `drawing_info` for a `Part_version`. A `drawing_info` need not be specified for a particular `Part_version`. See 4.3.347 for the application assertion.

#### **4.2.217.4 manufacture\_authorization**

The `manufacture_authorization` specifies the approval description for the manufacture of a `Part_version`. The `manufacture_authorization` need not be specified for a particular `Part_version`. There may be more than one `manufacture_authorization` for a `Part_version`. See 4.3.345 for the application assertion.

#### **4.2.217.5 manufactured\_by**

The `manufactured_by` specifies the person or organization responsible for the manufacture of the `Part_version`. There may be more than one `manufactured_by` for a `Part_version`. See 4.3.351 or 4.3.353 for the application assertion.

#### **4.2.217.6 material\_definition**

The `material_definition` specifies primary material choice of raw stock for producing the `Part_version`. There may be more than one `material_definition` for a `Part_version`. See 4.3.349 for the application assertion.



#### **4.2.217.7 owned\_by**

The `owned_by` specifies the person or organization that will own the completed manufactured `Part_version`. There may be more than one `owned_by` for a `Part_version`. See 4.3.351 or 4.3.353 for the application assertion.

#### **4.2.217.8 part\_description**

The `part_description` specifies in a textual format, human interpretable summary of the Part's characteristics which is appropriate for the manufacturing of the `Part_version`. Any comments or requirements on part characteristics may be given as a portion of the `part_description`.

#### **4.2.217.9 part\_id**

The `part_id` specifies a unique identifier for a `Part_version` within an organization.

#### **4.2.217.10 part\_name**

The `part_name` specifies a word or group of words by which a `Part_version` commonly is called within an organization. This name need not be unique within an organization.

#### **4.2.217.11 part\_revision\_id**

The `part_revision_id` specifies a unique identifier that defines the appropriate level of change that is incorporated into the design of the `Part_version` for the manufacturing function. The `part_revision_id` need not specify the latest version of a Part's design to be manufactured.

#### **4.2.217.12 physical\_form**

The `physical_form` specifies the shape of the `Part_version`. See 4.3.344 for the application assertion.

#### **4.2.217.13 property\_characteristic**

The `property_characteristic` specifies the associated data about the physical structure. The `property_characteristic` need not be specified for a particular `Part_version`. There may be more than one `property_characteristic` for a `Part_version`. See 4.3.355 for the application assertion.

#### **4.2.217.14 quantity\_ordered**

The `quantity_ordered` specifies the number of parts to be manufactured. There may be more than one `quantity_ordered` for a `Part_version`. See 4.3.350 for the application assertion.

### **4.2.217.15 security\_classification**

The `security_classification` specifies an organizational, national or international code that defines the availability of the `Part_version` or information about the `Part_version` to a particular individual, group of individuals, organization or group of organizations.

### **4.2.218 Parting\_surface**

A `Parting_surface` is a type of `Shape_element` (see 4.2.285) that is a joint of a specific contour or a plane of separation in a mould made of two or more section.

NOTE 1 Normally a `Parting_surface` is a planar surface, but it may be curved, or consist of several planar sections joined at angles.

NOTE 2 This application object is harmonized with the `parting_surface` object from ISO 8062-1:2007, Clause 4.13.

The data associated with a `Parting_surface` are the following:

- `cope_or_drag`;
- `male_or_female`;
- `offset_distance`;
- `shape_element`.

#### **4.2.218.1 cope\_or\_drag**

The `cope_or_drag` is a switch that specifies which side of the `parting_surface` is the `cope_side` surface or is the `drag_side` surface.

#### **4.2.218.2 offset\_distance**

An `offset_distance` specifies a dimension value for the amount of separation for the offset. See 4.3.357 for the application assertion.

#### **4.2.218.3 male\_or\_female**

A `male_or_female` is a switch that specifies which side of the `parting_surface` is the `cope_side` surface or is the `male` or `female` side.

#### **4.2.218.4 shape\_element**

The `shape_element` specifies the geometric form of the `cope` or `drag` `parting_surface`. See 4.3.356 and 4.3.358 for the application assertion.

### 4.2.219 Path\_element

A Path\_element is a type of Shape\_element (see 4.2.285) that is a continuous set of geometric curve that represent the path for a particular Machining\_feature.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.164.

### 4.2.220 Pattern\_equipment\_available

A Pattern\_equipment\_available is a type of Customer\_casting\_requirement (see 4.2.62) that declares the customer has pattern tooling and the tooling configuration. The data associated with a Pattern\_equipment\_available are the following:

- condition;
- core\_box\_cavity\_id;
- core\_box\_id;
- core\_box\_material;
- core\_per\_core\_box\_per\_assembly;
- flaskless\_pattern;
- flask\_size\_for\_mounted\_patterns;
- flask\_size\_for\_plate\_patterns;
- number\_of\_cores\_per\_casting;
- number\_of\_cores\_per\_box;
- number\_of\_core\_boxes;
- number\_of\_patterns;
- pattern;
- pattern\_impression\_id;
- pattern\_material;
- tooling\_location.

### **4.2.220.1 condition**

The condition specifies state of the pattern tooling.

NOTE See 4.2.220.1.1 to 4.2.220.1.4 for the definition of each allowable value for condition type.

**4.2.220.1.1 good:** the pattern tooling is in excellent shape no repair required.

**4.2.220.1.2 bad:** the pattern tooling is in bad need of repair or replacement.

**4.2.220.1.3 fair:** the pattern tooling is usable but needs some repair.

**4.2.220.1.4 new:** the pattern tooling has never been used.

### **4.2.220.2 core\_box\_cavity\_id**

The `core_box_cavity_id` specifies the identification for each cavity or impression of a core box when there is more than one cavity. There may be more than one `core_box_cavity_id` for a `Pattern_equipmnt_availability`. The `core_box_cavity_id` need not be specified for a particular `Pattern_equipmnt_availability`.

### **4.2.220.3 core\_box\_id**

The `core_box_id` specifies the alphanumeric identification for a core box. There may be more than one `core_box_id` for a `Pattern_equipmnt_availability`. The `core_box_id` need not be specified for a particular `Pattern_equipmnt_availability`.

### **4.2.220.4 core\_box\_material**

The core box material specifies the substance used to fabricate each core box required for the cast part. The `core_box_material` need not be specified for a particular `Pattern_equipmnt_availability` for the cast part.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for material type

### **4.2.220.5 core\_per\_core\_box\_per\_assembly**

The `core_per_core_box_per_assembly` specifies the cores for the casting to be from the same core box or from different core boxes. The `core_per_core_box_per_assembly` need not be specified for a particular `Pattern_equipmnt_availability` for the cast part.

NOTE See 4.2.220.5.1 to 4.2.220.5.2 for the definition of each allowable value for `core_per_core_box_per_assembly` type.

**4.2.220.5.1 same\_core:** all cores for the cast part are made from the same core box.

**4.2.220.5.2 different\_core:** cores for the cast part are made from different core boxes.

#### **4.2.220.6 flaskless\_pattern**

A `flaskless_pattern` is a pattern that is mounted in a wooden box or metal tub with a bottom that, when filled with no-bake sand, forms the mould for the casting. The `flaskless_pattern` need not be specified for a particular `Pattern_equipment_availability` for the cast part. See 4.3.360 for the application assertion.

Note 1 The mould is stripped from the box or tub and no mould jackets are required for this mould for pouring operations. Mould size is designed to withstand pouring pressures when mould is filled with molten metal.

NOTE 2 Since no flask pins are used the mould boxes are designed with a cope and drag matching system.

#### **4.2.220.7 flask\_size\_for\_mounted\_patterns**

The `flask_size_for_mounted_patterns` specifies the flask requirements to work with the pattern tooling as currently mounted on wooden boards to produce a cast part. The `flask_size_for_mounted_patterns` need not be specified for a particular `Pattern_equipment_availability` for the cast part. See 4.3.359 for the application assertion.

NOTE wooden boards included cope and drag boards, cope and drag insert boards, and matchboards or snapboard that require flask pin matching system.

#### **4.2.220.8 flask\_size\_for\_plate\_patterns**

The `flask_size_for_plate_patterns` specifies the flask requirements to work with the pattern tooling as currently mounted on metal plates to produce a cast part. The `flask_size_for_plate_patterns` need not be specified for a particular `Pattern_equipment_availability` for the cast part. See 4.3.359 for the application assertion.

NOTE metal plates included cope and drag plates, cope and drag insert plates, and matchplates that require flask pin matching system.

#### **4.2.220.9 number\_of\_cores\_per\_casting**

The `number_of_cores_per_casting` specifies the total number of cores required to produce the cast part. The `number_of_cores_per_casting` need not be specified for a particular `Pattern_equipment_availability` for the cast part.

#### **4.2.220.10 number\_of\_cores\_per\_box**

The `number_of_cores_per_box` specifies the total number of core cavities in the core box for the cast part. The `number_of_cores_per_box` need not be specified for a particular `Pattern_equipment_availability` for the cast part.

#### **4.2.220.11 number\_of\_core\_boxes**

The `number_of_core_boxes` specifies the total number of core boxes required to produce the cast part.

#### **4.2.220.12 number\_of\_patterns**

The `number_of_patterns` specifies an integer number for the quantity of pattern(s) on a mould board or matchplate.

#### **4.2.220.13 pattern**

The `pattern` specifies the type of impression shape used to form the mould cavity for the cast part.

NOTE See 4.2.220.13.1 to 4.2.220.13.11 for the definition of each allowable value for pattern type.

**4.2.220.13.1 loose\_pattern\_with\_follow\_on\_board:** a solid pattern that can not be mounted on a pattern board, but has a board that is hollowed out as pattern bed such that a natural parting line is formed so the drag and cope can be easily moulded.

**4.2.220.13.2 loose\_solid\_pattern:** a solid pattern that can not be mounted on a pattern board.

**4.2.220.13.3 loose\_split\_pattern:** a split pattern that can be mounted on a pattern board or plate.

**4.2.220.13.4 lost\_foam:** a pattern made from expandable polystyrene (EPS) foam or an alternative foam material.

**4.2.220.13.5 mounted\_cope\_and\_drag\_boards:** the cope half of the pattern is secured on a separate board to form the cope mould and the drag half secured on a separate board to form the drag mould. Each pattern half shall be matched to the pin centers of the board for the required flask.

NOTE boards could of wood, metal plates, or plastic

**4.2.220.13.6 mounted\_cope\_and\_drag\_flaskless:** the cope half of the pattern is secured on a separate board that is attached to a wooden box or metal tub to form the cope mould and the drag half of the pattern is secured on a separate board that is attached to a wooden box or metal tub to form the drag mould. Each pattern half shall be matched through some register mechanism as there are no flask matching pins with this moulding process.

NOTE boards could wooden, metal plates, or plastic

**4.2.220.13.7 mounted\_cope\_and\_drag\_inserts:** the cope half of the pattern is secured on a separate board of predetermined size, the drag half secured on a separate board of predetermined size. Each insert board is attached to an automatic moulding machine.

NOTE boards could of wood, metal plates, or plastic

**4.2.220.13.8 mounted\_match\_plate:** cope and drag pattern(s) are secured, back to back split along the parting line, to a metalplate.

**4.2.220.13.9 mounted\_snapboard:** cope and drag pattern(s) are secured, back to back split along the parting line, to a plywood board.

NOTE This pattern type will be for the squeezer or snapline line moulding process.

**4.2.220.13.10 shell pattern:** cope and drag metal pattern(s) are secured to a metal plate with a predetermined register system for matching the cope and drag moulds for the molten pouring process.

NOTE metal pattern are of cast iron, steel, and sometimes aluminum.

**4.2.220.13.11 sweep\_pattern:** a pattern that is made by the use of a template formed to a required shape or profile that is moved along a specified path to sweep a form for the pattern shape.

#### **4.2.220.14 pattern\_impression\_id**

The `pattern_impression_id` specifies the identification for each pattern impression on a mould board or matchplate when there is more than one pattern. There may be more than one `pattern_impression_id` for a `Pattern_equipment_availability`. The `pattern_impression_id` need not be specified for a particular `Pattern_equipment_availability`.

#### **4.2.220.15 pattern\_material**

The `pattern_material` specifies the substance used to produce the pattern.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for `core_box_material` type

#### **4.2.220.16 tooling\_location**

The `tooling_location` specifies where the pattern equipment is stored. See 4.3.361 and 4.3.362 for the application assertion.

#### **4.2.221 Pattern\_plate**

A `Pattern_plate` is a type of `Sand_casting_design_feature` (see 4.2.276) that is a metal plate or wooden board that one or more patterns and a gating system are mounted to, that establishes a parting surface for the cope and for the drag sections of a mould.

NOTE 1 A `Pattern_plate` can be a single plate or board. Or a `Pattern_plate` can be a pair of plates or boards, one for the cope side of the mould and one for the drag side of the mould. A `Pattern_plate` is usually a rectangular plate made of wood or metal.

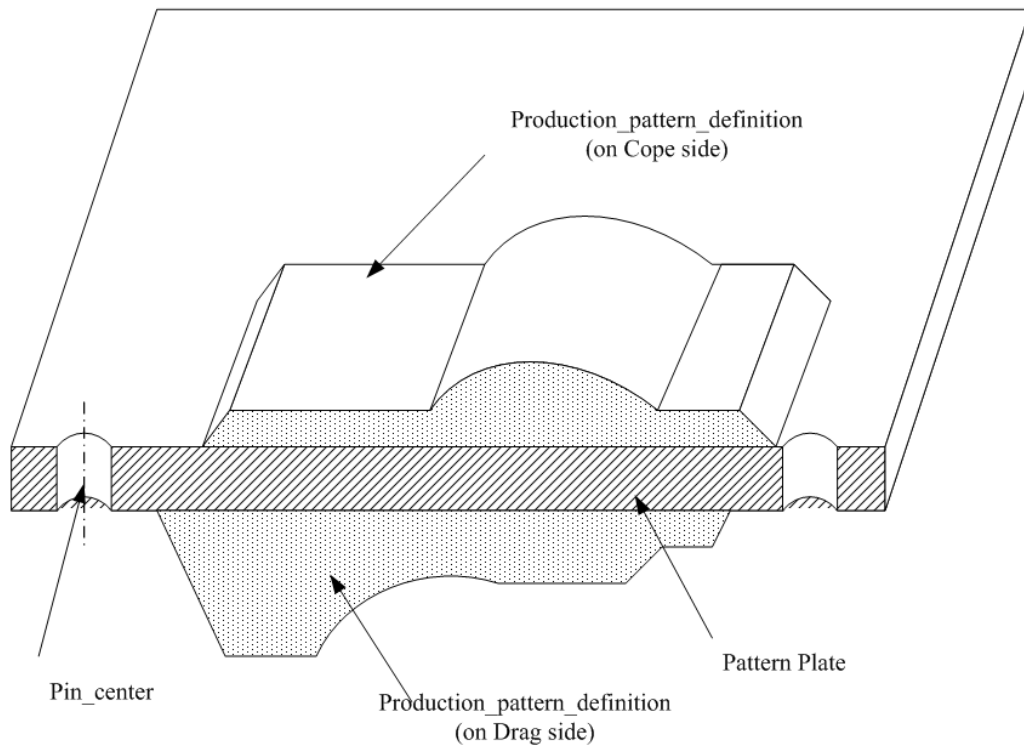
NOTE 2 Figure 15 illustrates the `Pattern_plate` with the `Production_pattern` split along parting line.

NOTE 3 This application object is harmonized with the matchplate definition from glossary of metalcasting terms from AFS [2].

ISO 10303-223:2008 (E)

The data associated with a Pattern\_plate are the following:

- defines\_flask\_type;
- gate\_rigging;
- lock\_definitions;
- mounting\_hole\_locations;
- moulding\_machine;
- parting\_line;
- pattern\_plate\_dimensions;
- pin\_center\_definition;
- type\_of\_plate.



**Figure 15 — Pattern\_plate**



#### **4.2.221.1 defines\_flask\_type**

The `defines_flask_type` specifies the flask size, pin centers and description for the flask required for the production pattern to cast the part. See 4.3.363 for the application assertion.

#### **4.2.221.2 gate\_rigging**

The `gate_rigging` specifies the gating system built onto the pattern plate. See 4.3.364 for the application assertion.

#### **4.2.221.3 lock\_definitions**

The `lock_definition` defines the method used to prevent mismatch of the casting when the cope and drag moulds are closed. See 4.3.367 for the application assertion.

#### **4.2.221.4 moulding\_machine**

The `moulding_machine` specifies the machine to which the pattern plate or board is attached to produce the mould. See 4.3.368 for the application assertion.

#### **4.2.221.5 mounting\_hole\_locations**

The `mounting_hole_locations` specifies the location of one or more holes required to secure the `Pattern_plate` to a moulding machine. The `machine_mounting_holes` need not be specified for a particular `Pattern_plate` for a cast part. See 4.3.366 for the application assertion.

#### **4.2.221.6 parting\_line**

The `parting_line` is a line around the pattern that defines the surfaces that separate the cope side of the pattern or mould and the drag side of the pattern or mould. See 4.3.369 for the application assertion.

#### **4.2.221.7 pattern\_plate\_dimensions**

The `pattern_plate_dimensions` specifies the implicit or explicit shape of the pattern plate. See 4.3.365 for the application assertion.

#### **4.2.221.8 pin\_center\_definition**

The `pin_center_definition` specifies the pin size and dimensional location for the centers of the pins that registers the mould when closed. The `pin_center_definition` need not be specified for a particular `Pattern_plate` for a cast part. See 4.3.370 for the application assertion.

NOTE there are many styles and kinds of plates or boards used to attach a pattern.

EXAMPLE `matchplate`, `snapboard`, `cope and drag inserts`, `cope and drag plates`, `moulding board` are some examples of common names for pattern plates.

#### **4.2.221.9 type\_of\_plate**

The `type_of_plate` specifies the kind of `Pattern_plate`.

EXAMPLE "`Pressure plate`" and "`match plate`" are two kinds of pattern plates.

#### **4.2.222 Pattern\_tooling\_requisition**

A `Pattern_tooling_requisition` is a type of `Requisition` (see 4.2.265) that describes the pattern equipment and tooling necessary to form the part as a casting during the manufacturing process.

#### **4.2.223 Payment\_and\_shipping**

The `Payment_and_shipping` is a type of `Customer_casting_requirement` (see 4.2.62) that defines the method of and requirements for payment. `Payment_and_shipping` also designates the method of shipping the cast parts to the customer.

The data associated with a `Payment_and_shipping` are the following:

- `customer_payment`;
- `packaging_or_creating`;
- `ship_casting`.

##### **4.2.223.1 customer\_payment**

The `customer_payment` specifies the conditions for payment of the cast part(s).

##### **4.2.223.2 packaging\_or\_creating**

The `packaging_or_creating` specifies the need for special handling of the cast part and describes textually the packaging or creating requirement. The `packaging_or_creating` need not be specified for a particular a `Payment_and_shipping` for the cast part.

### 4.2.223.3 **ship\_casting**

The `ship_casting` specifies the method to be used for shipping the product to the customer.

NOTE See 4.2.223.3.1 to 4.2.223.3.9 for the definition of each allowable value for `ship_casting` type.

**4.2.223.3.1 best way:** shipments made most economical or best delivery schedule.

**4.2.223.3.2 commercial air:** any airline company that provides commerce shipping for the public.

**4.2.223.3.3 commercial trucking:** any trucking company or common carrier that provides commerce shipping for the public.

**4.2.223.3.4 customer pickup:** the customer will provide their own shipping either their own truck or other means.

**4.2.223.3.5 FedEx:** shipments made by Federal Express.

**4.2.223.3.6 foundry trucking:** the foundry provides their own delivery service to the customer.

**4.2.223.3.7 rail:** any railway company that provides commerce shipping.

**4.2.223.3.8 UPS:** shipments made by United Parcel Service.

**4.2.223.3.9 other:** shipments made other ways.

### 4.2.224 **Pedigree\_creation\_order**

A `Pedigree_creation_order` is a document that defines the requirements for data to be gathered, documented and archived about a particular instance of a Part or batch of Parts throughout the manufacturing process.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.156.

The data associated with a `Pedigree_creation_order` are the following:

— `order_id`.

#### 4.2.224.1 **order\_id**

The `order_id` specifies a unique identifier for the `Pedigree_creation_order`.

### **4.2.225 Performance\_rate**

A `Performance_rate` is the time it takes to perform an `Activity`. Each `Performance_rate` may be either an `Allowed_time` (see 4.2.5) or a `Production_rate` (see 4.2.248).

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.111.

The data associated with a `Performance_rate` is the following:

- `source`.

#### **4.2.225.1 source**

The `source` specifies the origin of the `Performance_rate`, such as standards, policies and procedures, or estimates.

### **4.2.226 Permanent\_moulding\_process**

A `Permanent_moulding_process` is a type of `Casting_activity` (see 4.2.21) that is a process where the mould is used repeatedly many times to produce metal castings and for high production requirements. The data associated with a `Permanent_moulding_process` is the following:

- `ejection_definition`;
- `number_of_impressions`;
- `metal_flow_system`;
- `moulding_process`;
- `permanent_die_definition`;
- `vent_definition`.

#### **4.2.226.1 ejection\_definition**

The `ejection_definition` defines the ejector plate size and associated pins required for the `Ejector_system`. See 4.3.372 for the application assertion.

#### **4.2.226.2 number\_of\_impressions**

The `number_of_impressions` specifies the total number of die cavities in the production mould.

### 4.2.226.3 metal\_flow\_system

The metal\_flow\_system specifies each item of a gating system with their requirements that are needed to produce a cast part from a production permanent mould. See 4.3.375, 4.3.377, 4.3.373 for the application assertion.

### 4.2.226.4 moulding\_process

The moulding\_process specifies the system operations used to produce cast parts with a permanent mould.

NOTE See 4.2.226.4.1 to 4.2.226.4.4 for the definition of each allowable value for moulding\_process type.

**4.2.226.4.1 static:** the molten metal is poured into a mould sprue pouring cup in a vertical position. The mould is filled with the head pressure of the sprue.

**4.2.226.4.2 tilt:** the mould and the sprue pouring cup is filled in the horizontal position and is then rotated into the vertical position to fill the mould cavities.

**4.2.226.4.3 low\_pressure:** Low pressure air (2 to 5 psi) is applied to the molten metal chamber of the furnace that forces the molten metal into the mould cavity.

**4.2.226.4.4 vacuum chamber:** a vacuum is applied to thin sections of a casting being poured to aid in filling the die cavity of the mould.

### 4.2.226.5 permanent\_die\_definition

A permanent\_die\_definition defines the requirements for the two halves of a Production\_die\_mould to produce a cast part. See 4.3.374 for the application assertion.

### 4.2.226.6 vent\_definition

A vent\_definition defines the requirements to remove trapped air from the two halves of a production die mould as molten metal fills the die cavity of the mould. See 4.3.371 for the application assertion.

### 4.2.227 Person

A Person is an individual human being.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.167.

ISO 10303-223:2008 (E)

The data associated with a Person are the following:

- address;
- person\_id;
- person\_name;
- person\_phone\_number;
- title.

#### **4.2.227.1 address**

The address specifies a location at which a Person is contacted. This location may be physical or electronic. See 4.3.377 for the application assertion.

#### **4.2.227.2 person\_id**

The person\_id specifies the unique identification of an individual within the context to which that individual is being referenced.

EXAMPLE Within the United States individuals have unique social security numbers. Within a particular organization employing individuals, each individual may have a unique employee id.

#### **4.2.227.3 person\_name**

The person\_name specifies a word or group of words by which a Person is commonly referred. The person\_name may be comprised of compound elements that are commonly referred to as first name, last name, middle names and titles or suffixes.

#### **4.2.227.4 person\_phone\_number**

The person\_phone\_number specifies the telephone number at which the Person can be reached.

#### **4.2.227.5 title**

The title specifies a form of nomenclature indicating a persons status of responsibility in an organization or corporation. The title need not be specified for a particular Person.

### 4.2.228 Person\_in\_organization

A Person\_in\_organization is a person within the context of an organization.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.168.

The data associated with a Person\_in\_organization are the following:

- company;
- employee;
- role.

#### 4.2.228.1 company

The company specifies the functional group that employs the Person. See 4.3.378 for the application assertion.

#### 4.2.228.2 employee

The employee specifies Person employed by the Organization. See 4.3.379 for the application assertion.

#### 4.2.228.3 role

The role specifies the purpose that the person is fulfilling in an organization.

### 4.2.229 Pin\_center

A Pin\_center is a type of Sand\_casting\_design\_feature (see 4.2.276) that is the distance between the centers of two pins located on a centerline.

NOTE The pin centers of the flask are used as the pin centers for the pattern plate. This is what aligns the mould once the pattern plate has been removed and the mould is closed.

The data associated with a Pin\_center are the following:

- dimension.

#### 4.2.229.1 dimension

The dimension specifies a toleranced dimensional value for the distance between two pins. See 4.3.380 for the application assertion.

### **4.2.230 Pin\_tip**

The `Pin_tip` is a type of `Ejector_design_feature` (see 4.2.102) that is end of an ejector pin that intersects with the core cavity of the core box that forms the core. The data associated with a `Pin_tips` the following:

- `shape`.

#### **4.2.230.1 shape**

The `shape` defines the form and configuration of the pin tip. See 4.3.381 for the application assertion.

### **4.2.231 Placed\_target**

A `Placed_target` is a type of `Datum_target` (see 4.2.71) that is the implicit definition of a `Datum_target`. A `Placed_target` is either a `Target_point` (see 4.2.321), `Target_line` (see 4.2.320), `Target_rectangle` (see 4.2.322), or a `Target_circle` (see 4.2.319).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.169.

The data associated with `Placed_target` are the following:

- `placement`.

#### **4.2.231.1 placement**

The `placement` specifies location and orientation for the implicit definitions of the types of `Placed_target`. See 4.3.382 for the application assertion.

### **4.2.232 Planar\_element**

A `Planar_element` is a type of `Shape_element` (see 4.2.285) that is a flat surface.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.170.

The data associated with a `Planar_element` are the following:

- `location`;
- `normal`.

#### **4.2.232.1 location**

The `location` specifies the position of the planar surface. See 4.3.384 for the application assertion.



### 4.2.232.2 normal

The normal specifies the vector which indicates the normal of a plane being defined for the planar surface. See 4.3.383 for the application assertion.

### 4.2.233 Planar\_rib

A Planar\_rib is a type of Rib (see 4.2.271) that have planar rib walls. The rib top may be planar or have some defined profile shape. The data associated with a Planar\_rib the following:

- part\_rib\_radius;
- rib\_extent\_fillet;
- top;
- transition\_to\_top\_radius;
- walls;
- wall\_to\_floor\_radius;
- wall\_to\_top\_radius;

#### 4.2.233.1 part\_rib\_radius

The part\_rib\_radius specifies the dimensional value for the radius of a rib end that does not attach to any surface of the cast part. See 4.3.385 for the application assertion.

NOTE A part\_rib\_radius is not defined as a rib top.

#### 4.2.233.2 rib\_extent\_fillet

The rib\_extent\_fillet specifies the dimensional value for the transitional fillet radius between the rib side and the surface at the rib end(s) that intersects with any cast part surface that is not a rib floor. See 4.3.385 for the application assertion.

EXAMPLE The transition of a rib could be the rib end that intersects with a flange on the cast part.

#### 4.2.233.3 top

The top specifies one or more surface elements for the rib top. See 4.3.386 and 4.3.387 for the application assertion.

#### **4.2.233.4 transition\_to\_top\_radius**

The `transition_to_top_radius` specifies the dimensional value for the transitional radius between the rib top and a casting body shape. See 4.3.385 for the application assertion.

NOTE The `casting_body_shape` can be any part of the casting that extends beyond the rib top.

#### **4.2.233.5 walls**

The `walls` specifies one of more planar side elements for the Rib. See 4.3.387 for the application assertion.

#### **4.2.233.6 wall\_to\_top\_radius**

The `wall_to_top_radius` specifies the dimensional value for the transitional radius between the rib side and the rib top. See 4.3.385 for the application assertion.

#### **4.2.233.7 wall\_to\_floor\_radius**

The `wall_to_floor_radius` specifies the dimensional value for the transitional fillet radius between the rib side and the rib floor. See 4.3.385 for the application assertion.

NOTE The rib floors is defined as the casting body shape opposite the rib top.

#### **4.2.234 Plus\_minus\_value**

The `Plus_minus_values` is the upper and lower limits or tolerance value applied directly to a dimension. When applied to a `Dimensional_tolerance`, the `dimensional_value` is the tolerance value. When applied to a `Numeric_parameter_with_tolerance`, the `parameter_value` is the tolerance value.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.176.

EXAMPLE An illustration of `Dimensional_tolerance` with a `Plus_minus_value` is  $10 +.005 / -.002$ .

The data associated with a `Plus_minus_value` are the following:

- `lower_limit`;
- `significant_digits`;
- `upper_limit`.

#### **4.2.234.1 lower\_limit**

The `lower_limit` specifies the low limit value.

#### **4.2.234.2 significant\_digits**

The `significant_digits` specifies the number of decimal places indicating the accuracy of a dimension or tolerance.

#### **4.2.234.3 upper\_limit**

The `upper_limit` specifies the high limit value.

#### **4.2.235 Process\_parameter\_record**

A `Process_parameter_record` is a type of `Activity_result_record` (see 4.2.3) that is a log of the value of a manufacturing variable during manufacture of a part. The data associated with a `Process parameter record` are the following:

- `actual_parameter`;
- `planned_parameter`;
- `recorded_usage`.

##### **4.2.235.1 actual\_parameter**

The `actual parameter` specifies the measured value of a manufacturing variable. See 4.3.388 for the application assertion.

##### **4.2.235.2 planned\_parameter**

The `planned parameter` specifies the expected value of a manufacturing variable. See 4.3.388 for the application assertion.

##### **4.2.235.3 recorded\_usage**

The `recorded_usage` specifies the casting activity that the record applies to. See 4.3.389 for the application assertion.

## **4.2.236 Process\_plan**

An `Process_plan` is the collection of instructions, revisions, manufacturing activities and processes, along with their sequence of execution, that is required to machine a specific part. The data associated with `Process_plan` are the following:

- activities;
- activity\_results;
- description;
- defines\_manufacturing\_for;
- process\_plan\_identifier.

### **4.2.236.1 activities**

The `activities` specifies operations or tasks to be executed in the production of or machining of a cast part. There may be more than one activities for a `Process_plan`. The activities need not be specified for a particular `Process_plan`. See 4.3.391 for the application assertion.

### **4.2.236.2 activity\_results**

The `activity_results` specifies the conditions or outcome following the execution of an operation or task. See 4.3.390 for the application assertion.

### **4.2.236.3 defines\_manufacturing\_for**

The `defines_manufacturing_for` specifies the `Part_version` (see 4.2.217) that the process plan is being applied to. See 4.3.391 for the application assertion.

### **4.2.236.4 description**

The `description` specifies the word, or group of words, that relate detail information about the `Process_plan_version`.

### **4.2.236.5 process\_plan\_identifier**

The `process_plan_identifier` specifies an unique alphanumeric string that identifies the `Process_plan` used to produce the cast part.

### 4.2.237 **Process\_plan\_security**

A `Process_plan_security` is the level of protection that is to be exercised in the preparation, editing, modifying, and releasing the `Process_plan_version`. This security requirement is generated through contractual requirements with some governmental agency, or it is generated internally within an organization and deals with proprietary products. The data associated with `Process_plan_security` are the following:

- `classification_date`;
- `declassification_date`;
- `identified_by_activity`;
- `identified_by_process_plan`;
- `security_code`.

#### 4.2.237.1 **classification\_date**

The `classification_date` specifies the calendar date the security classification was granted.

#### 4.2.237.2 **declassification\_date**

The `declassification_date` specifies the calendar date the security classification was revoked or canceled.

#### 4.2.237.3 **identified\_by\_activity**

The `identified_by_activity` specifies the Activity (see 4.2.1) referenced by the `Process_plan_security` for the purpose of planning a machining activity. See 4.3.393 for the application assertion.

#### 4.2.237.4 **identified\_by\_process\_plan**

The `identified_by_process_plan` specifies the `Process_plan_version` (see 4.2.238) that requires security information. The `identified_by_process_plan` shall be specified for a particular `Process_plan_security`. See 4.3.394 for the application assertion.

#### 4.2.237.5 **security\_code**

The `security_code` specifies the level of security access that applies to the plan. It shall be equal to or higher than the highest `security_code` of any subsection within the `Process_plan_version`.

## **4.2.238 Process\_plan\_version**

An `Process_plan_version` is the collection of instructions, revisions, manufacturing activities and processes, along with their sequence of execution, that is required to machine a specific part.

The data associated with a `Process_plan_version` are the following:

- `activities_to_produce_part`;
- `amount_of_raw_materials`;
- `process_plan_information`;
- `process_plan_properties`;
- `quantity_of_parts`;

### **4.2.238.1 activities\_to\_produce\_part**

The `activities_to_produce_part` specifies the `Manufacturing_process` (see 4.2.180) that define the sequential list of activities controlled by the process plan. See 4.3.395 for the application assertion.

### **4.2.238.2 process\_plan\_information**

The `process_plan_information` specifies the `Special_instruction` (see 4.2.307) additional information for the process plan. A `process_plan_information` need not be specified for a particular `Process_plan_version`. See 4.3.399 for the application assertion.

### **4.2.238.3 process\_plan\_properties**

The `process_plan_properties` specifies the additional information about the process plan that is to be exchanged or archived. See 4.3.397 for the application assertion.

### **4.2.238.4 quantity\_of\_parts**

The `quantity_of_parts` specifies the count of parts to be manufactured per the process plan. A `quantity_of_parts` need not be specified for a particular `Process_plan_version`. See 4.3.397 for the application assertion.

### **4.2.238.5 amount\_of\_raw\_material**

The `amount_of_raw_material` specifies the `Material` (see 4.2.183) that is to be used to manufacture the product for this process plan. See 4.3.396 for the application assertion.

### 4.2.239 Process\_property

A Process\_property is the characteristics of a series of actions or operations directed toward changing the part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.180.

EXAMPLE Examples of Process\_property may be paint or coat.

The data associated with a Process\_property are the following:

- process\_name;
- property\_characteristic.

#### 4.2.239.1 process\_name

The process\_name specifies a word or group of words by which a Process\_property is commonly referred.

#### 4.2.239.2 property\_characteristic

The property\_characteristic specifies the parameter to describe the Process\_property. The property\_characteristic need not be specified for a particular Process\_property. There may be more than one property\_characteristic for a Process\_property. See 4.3.400 for the application assertion.

### 4.2.240 Process\_requirement

A Process\_requirement is a type of Customer\_casting\_requirement (see 4.2.62) that is an activity or operation that is required to produce the cast part. Each Process\_requirement may be a Heat\_treat\_requirement (see 4.2.145). The data associated with a Process\_requirement are the following:

- process\_name.

#### 4.2.240.1 process\_name

The process\_name is a descriptive title of the activity or operation for the process. See 4.3.401 for the application assertion.

EXAMPLE place chills, set chaplets, wash cores, inoculate ladle, remove gating, remove core sand, grind flash, load oven for heat treat are all examples of process names.

### **4.2.241 Product\_quality\_report**

A `Product_quality_report` is a type of `Document_assignment` (see 4.2.94) that is a description of a situation in which a requirement is not met. The data associated with a `Product_quality_report` are the following:

- `action_required`;
- `description`;
- `number_failed`;
- `what_failed`.

#### **4.2.241.1 action\_required**

An `action_required` specifies one or more textual messages describing corrective operations or changes to production procedures defined by the inspection plan.

#### **4.2.241.2 description**

The `description` specifies the textual account of the `Product_quality_report`. The `description` need not be specified for a particular `Product_quality_report`.

NOTE The `description` may include a list of the defects observed and reasons for production of scrap.

#### **4.2.241.3 number\_failed**

The `number_failed` specifies the quantity of `Master_sample` (see 4.2.182) objects that did not meet the requirements.

#### **4.2.241.4 what\_failed**

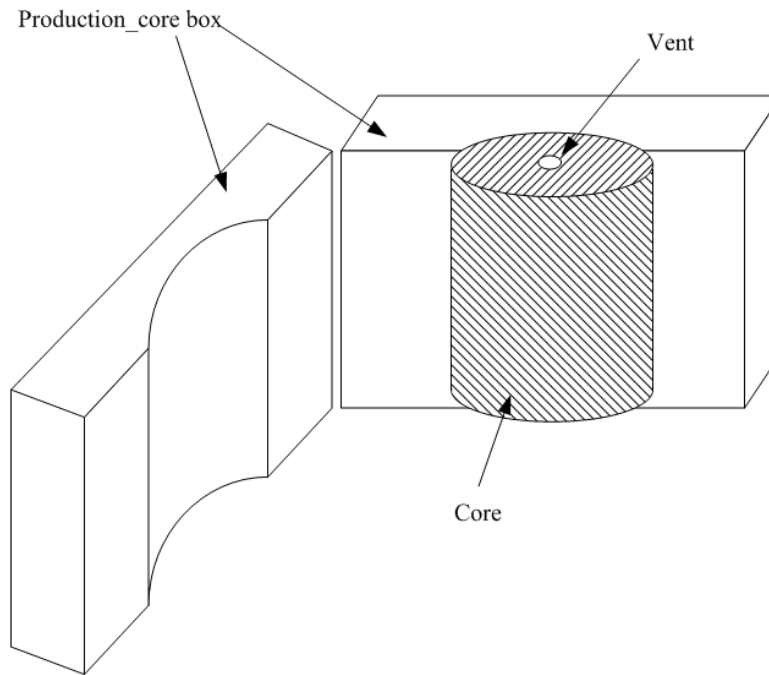
The `what_failed` specifies the `Inspection_or_test_result` (see 4.2.155) which gives details of what requirements the `Master_sample` (see 4.2.182) did not meet and how. See 4.3.402 for the application assertion.

### **4.2.242 Production\_core\_box**

A `Production_core_box` defines the structure and the specific shape of the desired core to be produced. The impression or cavity of the `Production_core_box` defines the female or negative shape of the desired core and will produce the male or positive core shape. Each `Production_core_box` may be a `Core_master` (see 4.2.56).

NOTE Figure 16 illustrates the `Production_core_box` with `Core`.





**Figure 16 — Production\_core\_box with Core**

The data associated with a Production\_core\_box are the following:

- box\_type;
- chaplet\_pad\_definition;
- cope\_and\_drag\_parting;
- core\_box\_stored;
- core\_box\_material;
- core\_definition;
- core\_made\_by;
- core\_print\_definition;
- draft;
- guide\_pin;

ISO 10303-223:2008 (E)

- impression\_shape;
- inserts;
- invest\_area;
- loose\_piece\_requirement;
- machine\_stock\_on\_tooling;
- number\_of\_impressions;
- offset\_parting\_surface;
- outside\_shape\_of\_core\_box;
- shrink\_factor;
- vent\_type.

#### **4.2.242.1 box\_type**

The `box_type` specifies a word or group of words by which the core box is commonly referred that makes the required core.

EXAMPLE split, dump, sweep, hot, shell, gang, etc. are examples of box types.

#### **4.2.242.2 chaplet\_pad\_definition**

The `chaplet_pad_definition` specifies the size and location of each `Chaplet_pad` attached to the core box. See 4.3.404 for the application assertion.

#### **4.2.242.3 cope\_and\_drag\_parting**

The `cope_and_drag_parting` is the information that defines the surface or surfaces that separates the cope side of the core box from the drag side of the core box. See 4.3.415 for the application assertion.

#### **4.2.242.4 core\_box\_stored**

The `core_box_stored` is the place of storage for the core box. See 4.3.409 and 4.3.410 for the application assertion.

#### **4.2.242.5 core\_box\_material**

The `core_box_material` specifies the material used to produce or fabricate the core box. Common materials are alum, cast iron, epoxy, hard wood, soft wood, steel, urethane.

**4.2.242.6 core\_definition**

The `core_definition` specifies one or more of the other geometric shapes and shape features for the core. See 4.3.405 for the application assertion.

**4.2.242.7 core\_made\_by**

The `core_made_by` specifies the process by which the core is produced. See 4.3.406 for the application assertion.

NOTE the core is either made manually as a bench job, semi-automatic by a conveyor system that may control cycle time, or is made by a core making machine.

**4.2.242.8 core\_print\_definition**

The `core_print_definition` specifies the size and shape of the core print with required clearance dimensions. See 4.3.407 for the application assertion.

**4.2.242.9 draft**

The `draft` specifies an angle in degrees of taper assigned to each vertical core surface that is not parallel with the Core box parting line. See 4.3.408 for the application assertion.

**4.2.242.10 guide\_pin**

The `guide_pin` specifies one or more pin diameters and length for the pins used to locate and register the mating halves of a core box. The `guide_pin` need not be specified for a particular `Production_core_box`. See 4.3.411 for the application assertion.

**4.2.242.11 impression\_shape**

The `impression_shape` defines the geometric shapes of the interior of the core box. See 4.3.418 for the application assertion.

NOTE when a core box has multiple impressions there will be multiple `impression_shape(s)` for the core box

**4.2.242.12 inserts**

The `inserts` specifies a secondary material that will become an integral part of the casting and where the insert is located. There may be more than one inserts for a `Production_core_box`. The inserts need not be specified for a particular `Production_core_box`. See 4.3.403 4.3.411 for the application assertion.

#### **4.2.242.13 invest\_area**

The `invest_area` specifies the the size and shape of the area in which the core making material is forced in the core cavity of the core box. See 4.3.418 for the application assertion.

#### **4.2.242.14 loose\_piece\_requirement**

The `loose_piece_requirement` specifies the geometric shapes of the loose piece. See 4.3.412 for the application assertion.

NOTE a loose piece may be required to mould an under cut, back draft, or a protrusion.

#### **4.2.242.15 machine\_stock\_on\_tool**

The `machine_stock_on_tool` specifies the amount of additional material that has been added to surfaces on the tool that shall be machined. There may be more than one `machine_stock_on_tool` for a `Production_core_box`. The `machine_stock_on_tool` need not be specified for a particular `Production_core_box`. See 4.3.413 for the application assertion.

#### **4.2.242.16 number\_of\_impressions**

The `number_of_impressions` specifies the number of core cavities in the core box.

NOTE Often referred to as a gang core box.

#### **4.2.242.17 offset\_parting\_surface**

The `offset_parting_surface` defines the required surfaces of the core box parting to allow the core to be removed freely. See 4.3.415 for the application assertion.

#### **4.2.242.18 outside\_shape\_of\_core\_box**

The `outside_shape_of_core_box` specifies the exterior surfaces of the core box. See 4.3.411 or 4.3.417 for the application assertion.

#### **4.2.242.19 shrink\_factor**

The `shrink_factor` specifies the rule that compensates for the casting shrinking in size as it cools to ambient temperatures.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE If the shrinkage compensation is one eighth inch per foot the shrink rule will be one eighth of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0104167 where the foot shrink rule will be 12.125 longer than the standard 12 inch foot rule.

#### 4.2.242.20 vent\_type

The vent\_type specifies the word or group of words states the kind of vent used in the core ore box. The vent\_type need not be specified for a particular Production\_core\_box. See 4.3.416 for the application assertion.

#### 4.2.243 Production\_die\_cast\_mould

A Production\_die\_cast\_mould is a type of Production\_tool (see 4.2.246) that defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces. A Production\_die\_cast\_mould may first be the definition for a Die\_cast\_master that will become the Production\_die\_cast\_mould.

NOTE When a Production\_die\_cast\_mould is cast to near net shape the Die\_cast\_master ( see 4.2.79) will be used to produce the Production\_die\_cast\_mould.

The data associated with a Production\_die\_cast\_mould are the following:

- cooling\_requirement;
- cooling\_type;
- die\_type;
- external\_box\_dimensions;
- gate\_runner;
- inserts;
- insert\_material\_type;
- internal\_shape;
- movable\_die\_items;
- port\_connection\_type;
- pouring\_sprue;
- secondary\_tooling\_requirements;
- stationary\_die\_items.

#### **4.2.243.1 cooling\_requirement**

The `cooling_requirement` specifies a boolean value that indicates if the production tooling requires cooling of the die mould to produce a casting. A value of true specifies that cooling is required. A value of false specifies that cooling is not required.

#### **4.2.243.2 cooling\_type**

The `cooling_type` specifies the requirements for one more cooling inlets and outlets for the production mould. A `cooling_type` need not be specified for a particular `Production_die_cast_mould`. See 4.3.419 for the application assertion.

#### **4.2.243.3 die\_type**

The `die_type` specifies the machine classification for which the die mould is made to produce a cast part.

NOTE See 4.2.243.3.1 to 4.2.243.3.3 for the definition of each allowable value for `die_type`.

**4.2.243.3.1 hot\_chamber:** a production mould that is designed and built to produce castings from a hot chamber diecasting machine.

**4.2.243.3.2 cold\_chamber:** a production mould that is designed and built to produce castings from a cold chamber diecasting machine.

**4.2.243.3.3 vacuum\_chamber:** a production mould that is designed and built to produce castings from a vacuum chamber investment casting machine.

#### **4.2.243.4 external\_box\_dimensions**

The `external_box_dimensions` specifies the maximum footprint of the `Production_die_cast_mould` that contains the ejector system and mounting to the diecasting machine. See 4.3.420 for the application assertion.

#### **4.2.243.5 gate\_runner**

The `gate_runner` specifies the ingate and runner for the die mould. There may be more than `gate_runner` when a combination die is used in the diecasting machine. See 4.3.423 for the application assertion.

#### **4.2.243.6 inserts**

An insert specifies the die mould that is inset into the die mould body when the die cast mould has been designed for multiple dies or removable dies. An insert need not be specified for a particular `Production_die_cast_mould`.

NOTE See 4.2.243.6.1 to 4.2.243.6.4 for the definition of each allowable value for insert type.

**4.2.243.6.1 unit:** several different dies from different customers are placed into a die cast holder for moulding with a common spruce to effect tooling and production economy.

**4.2.243.6.2 combination:** a set of dies that are placed into a die cast holder mould that is used to produce several parts for an assembly; sometimes called a family die mould.

**4.2.243.6.3 cast\_in:** a metallic object that is cast into the body of the casting which then becomes an integral part of the final casting.

**4.2.243.6.4 none:** no inserts are used for a Production\_die\_cast\_mould.

#### **4.2.243.7 insert\_material\_type**

The insert\_material\_type describes with words the required substance for the insert to be used. The insert\_material\_type need not be specified for a particular Production\_die\_cast\_mould.

#### **4.2.243.8 internal\_shape**

The internal\_shape defines the geometric shapes of the mould cavity of a die. See 4.3.425 for the application assertion.

NOTE When a die mould has multiple cavities there will be multiple cavity shapes for the die.

#### **4.2.243.9 movable\_die\_items**

The movable\_die\_item specifies one or more of the die items that move when the die mould is opened or closed. A movable\_die\_item need not be specified for a particular Production\_die\_cast\_mould. See 4.3.421 or 4.3.426 for the application assertion.

#### **4.2.243.10 port\_connection\_type**

A port\_connection\_type describes with words the style of cooling port connection required for the Production\_die\_cast\_mould.

#### **4.2.243.11 pouring\_sprue**

The pouring\_sprue specifies the requirements for connecting the nozzle of the diecasting machine with the runner in the die to fill the mould cavity. See 4.3.430 for the application assertion.

#### **4.2.243.12 secondary\_tooling\_requirements**

The stationary\_die\_item specifies one or more of the die items that are attached to the diecasting machine and does not move when the die mould is opened or closed. See 4.3.424 for the application assertion.

### **4.2.243.13 stationary\_die\_items**

The `stationary_die_item` specifies one or more of the die items that are attached to the diecasting machine and does not move when the die mould is opened or closed. See 4.3.421 and 4.3.422 for the application assertion.

### **4.2.244 Production\_die\_mould**

A `Production_die_mould` is a type of `Production_tool` (see 4.2.246) defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces. A `Production_die_mould` may first be the definition for a `Die_master` that will become the `Production_die_mould`.

NOTE When a `Production_die_mould` is cast to near net shape the `Die_master` ( see 4.2.84) will be used to produce the `Production_die_mould`.

The data associated with a `Production_die_mould` are the following:

- `core_for_die`;
- `die_type`;
- `gating`;
- `external_box_dimensions`;
- `slide_requirement`.

#### **4.2.244.1 core\_for\_die**

The `core_for_die` specifies the core or `die_core` required to make a casting with a die mould. See 4.3.427 and 4.3.428 for the application assertion.

#### **4.2.244.2 die\_type**

The `die_type` specifies the kind of die for the `Production_die_mould` to produce the cast part.

#### **4.2.244.3 external\_box\_dimensions**

The `external_box_dimension` specifies the maximum foot print of the `Production_die_mould` that contains the ejector system and mounting to moulding machine. See 4.3.431 for the application assertion.

#### **4.2.244.4 gating**

The `gating` specifies the gating system built onto the die mould. See 4.3.429 for the application assertion.



#### 4.2.244.5 **slide\_requirement**

The `slide_requirement` specifies the shape and material for one or more of the slides required to form one or more specific areas of a cast part. See 4.3.432 for the application assertion.

#### 4.2.245 **Production\_investment\_cast\_mould**

A `Production_investment_cast_mould` is a type of `Production_tool` (see 4.2.246). The `Production_investment_cast_mould` defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces. A `Production_investment_cast_mould` may first be the definition for an `Investment_casting_master` that will become the `Production_investment_cast_mould`. The data associated with a `Production_investment_cast_mould` are the following:

- `die_mould_storage`;
- `slide_requirement`.

##### 4.2.245.1 **die\_mould\_storage**

The `die_mould_storage` specifies the place of storage for the investment tooling in the facility. See 4.3.433 and 4.3.434 for the application assertion.

##### 4.2.245.2 **slide\_requirement**

The `slide_requirement` specifies the shape and material for a slide as part of the die mould that can only be shaped with the use of a slide to form a specific area of an investment casting. A `slide_requirement` need not be specified for a particular `slide_requirement`. See 4.3.435 for the application assertion.

#### 4.2.246 **Production\_tool**

The `Production_tool` is a mould which contain die cavities of the design shape required to produce production casting of a designed part for a customer. Each `Production_tool` is either a `Lost_foam_casting_die` (see 4.2.170), `Production_die_cast_mould` (see 4.2.243), `Production_die_mould` (see 4.2.244) or an `Production_investment_cast_mould` (see 4.2.245). The data associated with a `Production_tool` are the following:

- `allowance_on_machinable_surfaces`;
- `core_definition`;
- `core_for_die`;
- `draft`;
- `ejection_definition`;

ISO 10303-223:2008 (E)

- `guide_pin`;
- `male_or_female_mould`;
- `male_or_female_parting`;
- `mould_impression_shape`;
- `number_of_cavities`;
- `offset_parting_surfaces`;
- `riser_on_mould`;
- `vent_requirement`.

#### **4.2.246.1 allowance\_on\_machinable\_surfaces**

The `allowance_on_machinable_surfaces` specifies the stock to be added to a tool to permit machining. There may be more than one `Machining_allowance` specified for a particular die mould tool. See 4.3.440 for the application assertion.

NOTE When a Production mould is cast to near net shape, machine stock is added to the parting surface(s) and removable components for precision register, matches, and fits.

#### **4.2.246.2 core\_definition**

The `core_definition` specifies one or more of the other geometric shapes and shape features for the core. See 4.3.442 for the application assertion.

#### **4.2.246.3 core\_for\_die**

The `core_for_die` specifies one or more areas of the die impressions that require to form a section of the cast part. The `core_for_die` need not be specified for a particular `Production_tool`. See 4.3.436 for the application assertion.

NOTE When there are areas of the die mould that have undercuts, recesses, the interior part of the casting, or deep holes that can not be moulded as part of the die, then a core from a core box is required to mould or form that area of the cast part.

#### **4.2.246.4 draft**

The `draft` specifies an amount of taper or slope applied to a surface to permit the removal of the tool from the formed object in the tool. There may be more than one `draft` for a `Production_tool`. See 4.3.437 for the application assertion.

**4.2.246.5 ejection\_definition**

The `ejection_definition` defines the ejector plate size and associated pins required for the `Ejector_system`. See 4.3.438 for the application assertion.

**4.2.246.6 guide\_pin**

The `guide_pin` specifies one or more pin diameters and length for the pins used to located a production mould. See 4.3.439 for the application assertion.

**4.2.246.7 male\_or\_female\_mould**

The `male_or_female_mould` specifies a boolean value that indicates if the production mould is male or female. A value of true specifies that mould is male. A value of false specifies that mould is female.

**4.2.246.8 male\_or\_female\_parting**

The `male_or_female_parting` specifies one or more parting surfaces of a production mould for the male side of the mould or the female side of the mould. See 4.3.441 for the application assertion.

NOTE Theoretically the surface for the male parting is identical to the female parting surface and serve only to identify a specific section of the mould to produce the cast part. When the mould closes these two surfaces mate precisely to prevent runout or flash on the cast part at the parting line(s).

**4.2.246.9 mould\_impression\_shape**

The `mould_impression_shape` defines the geometric shapes of one or more mould cavity(s) of a production mould. See 4.3.446 for the application assertion.

**4.2.246.10 number\_of\_cavities**

The `numbers_of_cavities` specifies the total number of impressions in a production mould.

**4.2.246.11 offset\_parting\_surfaces**

An `offset_parting_surface` specifies one or more shapes for the parting when the parting surface is not a single plane. The `offset_parting_surface` need not be specified for a particular `Production_tool`. See 4.3.441 for the application assertion.

NOTE 1 An offset parting is required when the parting line of the die does not create a flat planar surface for the entire die mould including core prints.

NOTE 2 An offset parting may be two or more parallel planes offset by some specified distance and specified slope between the various parallel planes.

#### **4.2.246.12 outside\_shape\_of\_mould**

An `outside_shape_of_mould` specifies the exterior surfaces and profile of the `Production_tool`. See 4.3.439 and 4.3.445 for the application assertion.

#### **4.2.246.13 riser\_on\_mould**

The `riser_on_mould` specifies a reservoir of molten metal and necessary contact to the casting as part of the mould. There may be more than one `riser_on_mould` for a `Production_tool`. The `riser_on_mould` need not be specified for a particular `Production_tool`. See 4.3.443 and 4.3.444 for the application assertion.

#### **4.2.246.14 shrink\_factor**

The `shrink_factor` specifies the rule that compensates for the casting shrinking in size as it cools to ambient temperatures.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE If the shrinkage compensation is one eighth inch per foot the shrink rule will be one eighth of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0104167 where the foot shrink rule will be 12.125 longer than the standard 12 inch foot rule.

#### **4.2.246.15 vent\_requirement**

The `vent_requirement` specifies a need for a vent, its size, and where the vent is placed. There may be more than one `vent_requirement` for a `Production_tool`. The `vent_requirement` need not be specified for a particular `Production_tool`. See 4.3.447 for the application assertion.

#### **4.2.247 Production\_pattern\_definition**

The `Production_pattern_definition` defines the geometric shape, dimensional information, cores required, draft required, and parting surfaces. A `Production_pattern_definition` may first be the definition for a `Master_pattern` that will become the `Production_pattern`. The `Production_pattern_definition` may be a `Master_pattern` (see 4.2.181).

The data associated with a `Production_pattern_definition` are the following:

- `chaplet_pad_definition`;
- `chill_pattern`;
- `condition`
- `cope_and_drag_parting`;
- `cores_for_pattern`;

- core\_print\_definition;
- draft;
- inserts;
- loose\_piece\_requirement;
- number\_of\_patterns\_mounted;
- offset\_parting\_surface;
- pattern\_board\_size;
- pattern\_location;
- pattern\_shape;
- pattern\_with\_riser;
- shrink\_factor.

#### 4.2.247.1 chaplet\_pad\_definition

The chaplet\_pad\_definition specifies the size and location of each Chaplet\_pad attached to the Production\_pattern. See 4.3.449 for the application assertion.

#### 4.2.247.2 chill\_pattern

The chill\_pattern specifies the configuration of the external chill and the specific shape of the surface to be affected by the chill face. See 4.3.450 for the application assertion.

NOTE When an external chill is required to control solidification or hardness. The shape of the chill is to be exactly the shape of the cast part surface of the chilled area.

#### 4.2.247.3 condition

The condition specifies the state of the pattern tooling.

NOTE See 4.2.247.3.1 to 4.2.247.3.4 for the definition of each allowable value for condition type.

**4.2.247.3.1 good:** the pattern tooling is in excellent shape no repair required.

**4.2.247.3.2 bad:** the pattern tooling is in bad need of repair or replacement.

ISO 10303-223:2008 (E)

**4.2.247.3.3 fair:** the pattern tooling is usable but needs some repair.

**4.2.247.3.4 new:** the pattern tooling has never been used.

#### **4.2.247.4 cope\_and\_drag\_parting**

The cope and drag parting specifies the parting surface of the production pattern for the cope side of the mould and the drag side of the mould. See 4.3.456 for the application assertion.

NOTE Theoretically the surface for the cope parting is identical to the surface of the drag parting and serve only to identify a specific section of the mould to produce the cast part. When the mould closes these two surfaces match precisely to prevent runoff or flash on the cast part.

#### **4.2.247.5 cores\_for\_pattern**

The cores\_for\_pattern specifies the area(s) of the pattern that require a core to form that section of the cast part. See 4.3.451 for the application assertion.

NOTE when there are areas of the pattern that have undercuts, recesses, the interior part of the casting, or deep holes that can not be moulded as part of the pattern, then a core box is required to mould or form that area of the cast part.

#### **4.2.247.6 core\_print\_definition**

The core\_print\_definition specifies the parameters and form for the core print. See 4.3.452 for the application assertion.

NOTE A core print is require for each core that the pattern requires. A core print provides the seat for the core to rest in and to align or secure the core in the mould or in another core.

#### **4.2.247.7 draft**

The draft specifies an angle in degrees of slope or taper.

NOTE The draft is assigned to each vertical surface of the Production\_pattern that is not parallel with the parting line or surface of the pattern. For loose pieces of the pattern, the draft is applied in the direction that the loose piece is drawn from the sand mould. See 4.3.453 for the application assertion.

#### **4.2.247.8 inserts**

The inserts specifies a secondary material that will become an integral part of the casting and where the insert is located. There may be more than one inserts for a Production\_pattern\_definition. The inserts need not be specified for a particular Production\_pattern\_definition. See 4.3.448 for the application assertion.

#### **4.2.247.9 loose\_piece\_requirement**

The `loose_piece_requirement` specifies an area of a pattern that can be formed in the mould with a removeable part of the pattern. See 4.3.455 for the application assertion.

NOTE A loose piece is often used instead of a core to mould a specific area of the pattern that will not draw from the sand mould.

#### **4.2.247.10 number\_of\_patterns\_mounted**

The `number_patterns_mounted` specifies the number of patterns mounted on the pattern board.

#### **4.2.247.11 pattern\_board\_size**

The `pattern_board_size` the pattern plate or board used to make the cope parting surface and the drag parting surface of the mould. See 4.3.457 for the application assertion.

NOTE A pattern is secured a metal plate or wooden board at the parting line of a pattern. The pattern plate is larger than the size of the flask because the flask is to set on the pattern plate.

#### **4.2.247.12 offset\_parting\_surface**

An `offset_parting_surface` specifies the shape of the parting when the parting surface is not a single plane. See 4.3.456 for the application assertion.

NOTE 1 An offset parting is required when the parting line of the pattern does not create a flat planar surface for the entire pattern including core prints, i.e, a flat plate or board.

NOTE 2 An offset parting may be two or more parallel planes offset by some specified distance and specified slope between the two parallel planes.

#### **4.2.247.13 pattern\_location**

The `pattern_location` specifies the place of storage for the pattern in the facility. See 4.3.454 for the application assertion.

#### **4.2.247.14 pattern\_shape**

The `pattern_shape` specifies the outside geometrical shape and form with the dimensional values of the pattern. The `pattern_shape` will also specify the `loose_piece` attachment areas along with their geometric shape and dimensional values. See 4.3.458 for the application assertion.

#### **4.2.247.15 riser\_on\_pattern**

The `riser_on_pattern` specifies a reservoir requirement for molten metal and necessary contact pad on the pattern for the casting. There may be more than one `riser_on_pattern` for a `Production_pattern_definition`.

ISO 10303-223:2008 (E)

The `riser_on_pattern` need not be specified for a particular `Production_pattern_definition`. See 4.3.459 and 4.3.460 for the application assertion.

#### **4.2.247.15.1 shrink\_factor**

The `shrink_factor` specifies the rule that compensates for the casting shrinking in size as it cools to ambient temperatures.

NOTE A shrink rule is used by the patternmaker to scale the pattern for the shrinkage compensation.

EXAMPLE if the shrinkage compensation is one eighth inch per foot the shrink rule will be one eighth of an inch longer than a standard twelve inch foot rule. Or the shrink factor for the same foot rule is 1.0104167 where the foot shrink rule will be 12.125 longer than the standard 12 inch foot rule.

#### **4.2.248 Production\_rate**

A `Production_rate` is a type of `Performance_rate` (see 4.2.225) that is the time it takes to produce a measurable amount of items.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.118.

EXAMPLE `Production_rate` includes such things as machine five parts per hour or drill 3 holes per minute.

The data associated with a `Production_rate` are the following:

- `time_per_unit`;
- `unit_quantity`.

##### **4.2.248.1 time\_per\_unit**

The `time_per_unit` specifies the duration for completing one task.

##### **4.2.248.2 unit\_quantity**

The `unit_quantity` specifies the number of units to be produced.

#### **4.2.249 Production\_welding**

In `Production_welding` is a type of `Non_casting_operations` (see 4.2.202) that is a type of `Non_casting_` operation and an added value operation that joins two or more cast parts or a cast part to wrought materials by a welding procedure before any machining is performed.



## 4.2.250 Project\_order

A Project\_order is a document that is used to initiate and track a project within the manufacturing organization. A Project\_order may track a single part or a part batch.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.184.

The data associated with a Project\_order are the following:

- ordered\_resource;
- part\_status;
- pedigree\_creation\_status;
- project\_order\_id;
- release\_authorization;
- resource\_acquisition\_status;
- shop\_work\_status;
- technical\_data\_package\_status.

### 4.2.250.1 ordered\_resource

The ordered\_resource specifies the resources required by the Project\_order. There may be more than one ordered\_resource for a Project\_order. See 4.3.465 for the application assertion.

### 4.2.250.2 part\_status

The part\_status specifies part that is tracked by the Project\_order. There may be more than one part\_status for a Project\_order. See 4.3.463 for the application assertion.

### 4.2.250.3 pedigree\_creation\_status

The pedigree\_creation\_status specifies the Pedigree\_creation\_order that is tracked by the Project\_order. The pedigree\_creation\_status need not be specified for a particular Project\_order. See 4.3.464 for the application assertion.

### 4.2.250.4 project\_order\_id

The project\_order\_id specifies a unique identifier for the Project\_order.

#### **4.2.250.5 release\_authorization**

The `release_authorization` specifies the approval to manufacture a Part. See 4.3.461 for the application assertion.

#### **4.2.250.6 resource\_acquisition\_status**

The `resource_acquisition_status` specifies `Resource_acquisition_order` that is tracked by the `Project_order`. The `resource_acquisition_status` need not be specified for a particular `Project_order`. See 4.3.466 for the application assertion.

#### **4.2.250.7 shop\_work\_status**

The `shop_work_status` specifies the `shop_work_order` that is tracked by the `Project_order`. The `shop_work_status` need not be specified for a particular `Project_order`. See 4.3.467 for the application assertion.

#### **4.2.250.8 technical\_data\_package\_status**

The `technical_data_package_status` specifies the `Digital_technical_data_package_work_order` that is tracked by the `Project_order`. The `technical_data_package_status` need not be specified for a particular `Project_order`. See 4.3.462 for the application assertion.

### **4.2.251 Projection**

A Projection is an extension of a feature from the part so that a tolerance zone can be created. A feature is extended from one end of a feature for a specified length.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.185.

The data associated with a Projection are the following:

- `projection_end`;
- `projection_length`.

#### **4.2.251.1 projection\_end**

The `projection_end` specifies the physical shape that is the extension of a feature. See 4.3.468 for the application assertion.

#### **4.2.251.2 projection\_length**

The `projection_length` specifies the amount to extend the end of a feature.

### 4.2.252 Proof\_report

A Proof\_report is a type of Document\_assignment (see 4.2.94) that details the evidence of how the cores fit together when assembled and placed in the mould as required.

### 4.2.253 Property

A Property is a characteristic associated with the physical structure or integrity of an element of a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.186.

The data associated with a Property are the following:

- description;
- material\_characteristic;
- part\_characteristics;
- process\_characteristics;
- property\_description;
- property\_name;
- shape\_property\_characteristics;
- surface\_characteristics;
- weight\_characteristics.

#### 4.2.253.1 description

The description specifies the textual account of an Property.

#### 4.2.253.2 material\_characteristic

The material\_characteristic specifies the information that describe material for manufacturing the Part. The material\_characteristic need not be specified for a particular Property. There may be more than one material\_characteristic for a Property. See 4.3.470 for the application assertion.

#### 4.2.253.3 part\_characteristics

The part\_characteristic specifies the information that describe properties of the Part. The part\_characteristic need not be specified for a particular Property. There may be more than one part\_characteristic for a Property. See 4.3.471 for the application assertion.

#### **4.2.253.4 process\_characteristics**

The process\_characteristic specifies information that describe processes for manufacturing the part. The process\_characteristic need not be specified for a particular Property. There may be more than one process\_characteristic for a Property. See 4.3.472 for the application assertion.

#### **4.2.253.5 property\_description**

The property\_description specifies the Specification that has additional information about the properties of the Part. The property\_description need not be specified for a particular Property. There may be more than one property\_description for a Property. See 4.3.474 for the application assertion.

#### **4.2.253.6 property\_name**

The property\_name specifies a word or group of words by which a property is commonly referred.

#### **4.2.253.7 shape\_property\_characteristic**

The shape\_property\_characteristic specifies information that describe shape properties of the Part. The shape\_property\_characteristic need not be specified for a particular Property. There may be more than one shape\_property\_characteristic for a Property. See 4.3.473 for the application assertion.

#### **4.2.253.8 surface\_characteristics**

The surface\_characteristic specifies information that describe surface conditions of the Part. The surface\_characteristic need not be specified for a particular Property. There may be more than one surface\_characteristic for a Property. See 4.3.475 for the application assertion.

#### **4.2.253.9 weight\_characteristics**

The weight\_characteristics specifies a weight value with units for the cast part. The weight\_characteristics need not be specified for a particular Property. There may be more than one weight\_characteristic for a Property. See 4.3.469 for the application assertion.

#### **4.2.254 Property\_BSU**

A Property\_BSU is a type of BSU (see 4.2.18) that identifies a property basic semantical unit of a class in a parts library.

The data associated with a Property\_BSU are the following:

- name\_scope;
- version.

**4.2.254.1 name\_scope**

The `name_scope` specifies the class this property belongs to. See 4.3.476 for the application assertion.

**4.2.254.2 version**

The `version` specifies the designation of the version of the information piece.

**4.2.255 Property\_inspection**

The `Property_inspection` is a type of `Inspection_or_test_result` (see 4.2.155) that is the outcome of a measurement and comparing the resultant properties to the design requirement. The data associated with a `Property_inspection` are the following:

- `requirement`;
- `property_result`.

**4.2.255.1 property\_result**

The `property_result` specifies the outcome of the `Property_inspection` against the requirements. See 4.3.477 for the application assertion.

**4.2.255.2 requirement**

The `requirement` specifies the `Property_parameter` (see 4.2.256) that was evaluated to produce the result. See 4.3.477 for the application assertion.

**4.2.256 Property\_parameter**

A `Property_parameter` is an element of information that describes a characteristic that comprises the property. Each `Property_parameter` may be one of the following: `Descriptive_parameter` (see 4.2.74) or a `Numeric_parameter` (see 4.2.205).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.188.

The data associated with a `Property_parameter` are the following:

- `parameter_name`.

**4.2.256.1 parameter\_name**

The `parameter_name` specifies a word or group of words that identify a characteristic of interest for a `Property_parameter`.

### **4.2.257 Property\_relationship**

A `Property_relationship` is the allowable pairs of values for two variables.

EXAMPLE A plot of heat treating oven temperature versus time.

The data associated with a `Property_relationship` are the following:

- `relationship_curve`;
- `x_property_name`;
- `y_property_name`.

#### **4.2.257.1 relationship\_curve**

The `relationship_curve` specifies the x-y plot that is used to describe the correlation between the two variables. See 4.3.478 for the application assertion.

#### **4.2.257.2 x\_property\_name**

The `x_property_name` specifies the label used to identify the second property.

#### **4.2.257.3 y\_property\_name**

The `y_property_name` specifies the label used to identify the first property.

### **4.2.258 Property\_value**

A `Property_value` provides a value for a property as specified in the property basic semantical unit. The value type is specified in subtypes.

The data associated with a `Property_value` are the following:

- `property_BSU`;
- `value_amount`.

#### **4.2.258.1 property\_BSU**

The `property_BSU` specifies the `Property_BSU` (see 4.2.254) that defines the basic semantical unit. See 4.3.479 for the application assertion.

#### 4.2.258.2 value\_amount

The value\_amount specifies the value that is defined as a boolean, integer, number, logical, string, or real.

#### 4.2.259 Quality\_acceptance\_report

A Quality\_acceptance\_report is a type of Document\_assignment (see 4.2.94) that is a documented record of the inspection and test results for a cast part. The data associated with a Quality\_acceptance\_report are the following:

— applied\_to.

##### 4.2.259.1 applied\_to

The applied\_to specifies one or more quality reports for the cast part. See 4.3.480 for the application assertion.

#### 4.2.260 Radial\_dimension\_tolerance

A Radial\_dimension\_tolerance is a type of Size\_tolerance (see 4.2.304) that is the allowable variation for the radial distance from the center of a circular curve to a point on the curve.

NOTE 1 This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.191.

NOTE 2 Figure 17 illustrates the Radial\_dimension\_tolerance.

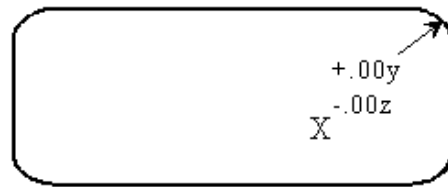


Figure 17 — Radial\_dimension\_tolerance

#### 4.2.261 Red\_line\_part

A Red\_line\_part is a type of Part\_version (see 4.2.217) that identifies design inconsistencies for the part version being manufactured and may require engineering design changes to produce the Part\_version (see 4.2.217) as a casting.

ISO 10303-223:2008 (E)

The data associated with a Red\_line\_part are the following:

- mark\_up\_feature;
- mark\_up\_shape;
- mark\_up\_material;
- mark\_up\_property.

#### **4.2.261.1 mark\_up\_feature**

The mark\_up\_feature specifies Feature (see 4.2.124) of the part that may require a design change. There may be more than one mark\_up\_feature for a Red\_line\_part. The mark\_up\_feature need not be specified for a particular Red\_line\_part. See 4.3.481 for the application assertion.

#### **4.2.261.2 mark\_up\_shape**

The mark\_up\_shape specifies Shape\_aspect (see 4.2.282) of the part that may require a design change. There may be more than one mark\_up\_shape for a Red\_line\_part. The mark\_up\_shape need not be specified for a particular Red\_line\_part. See 4.3.484 for the application assertion.

#### **4.2.261.3 mark\_up\_material**

The mark\_up\_material specifies Material (see 4.2.183) for the part that may require a design change. There may be more than one mark\_up\_material for a Red\_line\_part. The mark\_up\_material need not be specified for a particular Red\_line\_part. See 4.3.482 for the application assertion.

#### **4.2.261.4 mark\_up\_property**

The mark\_up\_property specifies Property (see 4.2.253) of the part that may require a design change. There may be more than one mark\_up\_property for a Red\_line\_part. The mark\_up\_property need not be specified for a particular Red\_line\_part. See 4.3.483 for the application assertion.

### **4.2.262 Reporting\_requirement**

A Reporting\_requirement is a type of Customer\_casting\_requirement (see 4.2.62) that is the information that shall be disclosed. The data associated with a Reporting\_requirement are the following:

- reporting\_based\_on;
- date\_to\_be\_reported.



### **4.2.262.1 reporting\_based\_on**

A `reporting_based_on` requirement specifies the standard, specification, and guideline that is to be followed and reported for a specific area of the cast part. See 4.3.486 for the application assertion.

### **4.2.262.2 date\_to\_be\_reported**

The `date_to_be_reported` specifies the day, month, and year the report is conducted or completed. See 4.3.485 for the application assertion.

## **4.2.263 Request\_for\_clarification**

A Request for clarification is an appeal for missing information, or additional information to eliminate possible sources of confusion or error concerning a design

EXAMPLE 1 A dimension that is not supplied.

EXAMPLE 2 Dimensions that do not add up.

The data associated with a `Request_for_clarification` are the following:

- `description`;
- `request_date`;
- `requested_by`;
- `requested_part`.

### **4.2.263.1 description**

The `description` specifies the textual account of a clarification request.

### **4.2.263.2 request\_date**

The `request_date` specifies the date and time at which the request is made. See 4.3.489 for the application assertion.

### **4.2.263.3 requested\_by**

The `requested_by` specifies the person and organization making the request. See 4.3.487 for the application assertion.

#### **4.2.263.4 requested\_part**

The requested\_part specifies the Part\_version (see 4.2.217) that is affected by the request. See 4.3.488 for the application assertion.

#### **4.2.264 Request\_for\_quotation**

A Request\_for\_quotation is a solicitation from a customer to a casting or tooling supplier to bid or quote on producing cast parts or tooling as per a design specification. The data associated with a Request\_for\_quotation are the following:

- delivery\_date;
- order\_completion;
- person\_submitting\_quote;
- quantity\_per\_month;
- quantity\_breaks;
- quantity\_per\_release;
- quantity\_per\_year;
- redline;
- reply\_due\_date;
- request\_date;
- RFQ\_number;
- total\_quantity\_per\_order;
- type\_of\_quotation.

##### **4.2.264.1 delivery\_date**

The delivery\_date specifies the month, day, and year the product shall be received by the customer. See 4.3.490 for the application assertion.

NOTE the customer may be the company buying the cast part or it could be the foundry requesting the pattern equipment for the cast part.

#### **4.2.264.2 order\_completion**

The `order_completion` specifies the the month, day, and year the customer order is to be completed and received by the customer. See 4.3.490 for the application assertion.

#### **4.2.264.3 person\_submitting\_quote**

The `person_submitting_quote` specifies an individual within the organization who is making or presenting the quote for the request made. There may be more than one `person_submitting_quote` for a particular `Request_for_quotation` for a cast part. See 4.3.491 for the application assertion.

#### **4.2.264.4 quantity\_per\_month**

The `quantity_per_month` specifies the number of cast parts to be delivered each month to the customer as long as the customer order is valid. The `quantity_per_month` need not be specified for a particular `Request_for_quotation`.

#### **4.2.264.5 redline**

The redline specifies a `Red_line_part` (see 4.2.261) with the quotation when submitted. The redline need not be specified for a particular `Request_for_quotation`. See 4.3.492 for the application assertion.

#### **4.2.264.6 reply\_due\_date**

The `reply_due_date` specifies the month, day, and year the complete `Request_for_quotation` is to be in the customer's possession. See 4.3.490 for the application assertion.

#### **4.2.264.7 RFQ\_number**

The `RFQ_number` specifies a unique alphanumeric string that will identify the customer's `Request_for_quotation`.

#### **4.2.264.8 request\_date**

The `reply_due_date` specifies the month, day, and year the complete `Request_for_quotation` is to be in the customer's possession. See 4.3.490 for the application assertion.

#### **4.2.264.9 quantity\_breaks**

The `quantity_break` specifies the number of cast parts for a specific quoted price. There may be more than one `quantity_break` for a particular `Request_for_quotation` for a cast part. See 4.3.490 for the application assertion.

#### **4.2.264.10 quantity\_per\_release**

The `quantity_per_release` specifies the number of cast parts to be delivered at a time to the customer as long as the customer order is valid. The `quantity_per_release` need not be specified for a particular `Request_for_quotation`. See 4.3.490 for the application assertion.

#### **4.2.264.11 quantity\_per\_year**

The `quantity_per_year` specifies the number or estimated number of cast parts to be delivered each year to the customer as long as the customer order is valid. The `quantity_per_year` need not be specified for a particular `Request_for_quotation`. See 4.3.490 for the application assertion.

#### **4.2.264.12 total\_quantity\_per\_order**

The `total_quantity_per_order` specifies the total number of peices to be produced for the customer order. See 4.3.490 for the application assertion.

#### **4.2.264.13 type\_of\_quotation**

The `type_of_quotation` specifies the kind of product that is requested to be quoted. See 4.3.490 for the application assertion.

EXAMPLE: the kinds of quotation could be for Die Castings, Investment Castings, Green sand Castings, Zinc Die Castings, or Die tooling are just examples of the possible types of quotes.

### **4.2.265 Requisition**

A Requisition is a document that contains an order for equipment or materials to perform or support the manufacturing process. Each Requisition may be one of the following: `Die_mould_requisition` (see 4.2.85), `Pattern_tooling_requisition` (see 4.2.222), `Material_requisition` (see 4.2.187) or an `Indirect_stock_requisition` (see 4.2.150).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.205.

The data associated with a Requisition are the following:

- `quantity_ordered`;
- `required_delivery_date`;
- `requisition_date`;
- `requisition_description`;
- `requisition_number`.

**4.2.265.1 quantity\_ordered**

The `quantity_ordered` specifies the number of items that are being purchased by an organization using the Requisition.

**4.2.265.2 required\_delivery\_date**

The `required_delivery_date` specifies the year, month and day on which the item or items specified in the Requisition are required at the site that ordered them. See 4.3.493 for the application assertion.

**4.2.265.3 requisition\_date**

The `requisition_date` specifies the year, month, and day on which the Requisition was issued. See 4.3.493 for the application assertion.

**4.2.265.4 requisition\_description**

The `requisition_description` specifies human interpretable prose that describes the item being ordered, and provides any special instructions that may pertain to the order.

**4.2.265.5 requisition\_number**

The `requisition_number` specifies a unique identification of the Requisition.

**4.2.266 Reset\_or\_return\_pins**

The `Reset_or_return_pins` is a type of `Ejector_design_feature` (see 4.2.102) that is the mechanism of pins that repositions the ejector plate to its initial position for the cycle. The data associated with a `Reset_or_return_pins` are the following:

- `pin_dimensions`;
- `spring_discription`.

**4.2.266.1 pin\_dimensions**

The `pin_dimensions` specifies the size measurement information for each reset or return pin used in the ejector system. See 4.3.494 for the application assertion.

### **4.2.266.2 spring\_discription**

The `spring_discription` describes the requirement characteristics of the spring. A `spring_discription` need not be specified for a particular `Reset_or_return_pins`.

NOTE the spring type will state the inside and outside diameter of the spring as well as the tension or compression load for the spring.

EXAMPLE of requirements would be: free overall length, solid overall length when compressed, wire diameter or gauge, inside or outside diameters, deflection force load or working stress load(psi), etc.

### **4.2.267 Resource\_acquisition\_order**

A `Resource_acquisition_order` is a document used for process planning of a cast part. This order defines the requirements for material needed to produce a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.206.

EXAMPLE The `Resource_acquisition_order` may contain customer order information, part identification, quantity, or schedules.

The data associated with a `Resource_acquisition_order` are the following:

— `order_id`.

#### **4.2.267.1 order\_id**

The `order_id` specifies a unique identifier for the `Resource_acquisition_order`.

### **4.2.268 Resource\_with\_material**

A `Resource_with_material` is a type of `Generic_casting_resource` (see 4.2.136) that provides the means to reference material information about a resource.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.125.

The data associated with a `Resource_with_material` are the following:

— `resource_material`.

#### **4.2.268.1 resource\_material**

The `resource_material` specifies the `Material` (see 4.2.183) that is required as a resource material to manufacture a part. See 4.3.495 for the application assertion.

### 4.2.269 Resource\_with\_representation

A Resource\_with\_representation is a type of Generic\_casting\_resource (see 4.2.136) that provides the means to reference information about a resource within a parts library dictionary.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.126.

The data associated with a Resource\_with\_representation are the following:

- resource\_documentaton.

#### 4.2.269.1 resource\_documentaton

The resource\_documentaton specifies the Externally\_defined\_representation (see 4.2.119). See 4.3.496 for the application assertion.

### 4.2.270 Revision

A Revision is the identification of a change that is incorporated into the process plan.

NOTE The revision history may be derived from the different instances of revision.

The data associated with Revision are the following:

- approved\_by;
- description;
- reason\_for\_revision;
- related\_to;
- relating\_to;
- revision\_level.

#### 4.2.270.1 approved\_by

The approved\_by specifies the Status\_authority (see 4.2.312) that defines approval of the revision. See 4.3.497 for the application assertion.

#### 4.2.270.2 description

The description specifies the word, or group of words, that explain the Revision.

### **4.2.270.3 reason\_for\_revision**

The `reason_for_revision` specifies the word, or group of words, that explain the need for the change.

### **4.2.270.4 related\_to**

The `related_to` specifies the `Process_plan_version` (see 4.2.238) that is the successor process plan. See 4.3.498 for the application assertion.

### **4.2.270.5 relating\_to**

The `relating_to` specifies the `Process_plan_version` (see 4.2.238) that is the predecessor process plan. See 4.3.498 for the application assertion.

### **4.2.270.6 revision\_level**

The `revision_level` specifies the unique identification, within a process plan or activity, of the change level.

### **4.2.271 Rib**

A `Rib` is a type of `Casting_design_feature` (see 4.2.22) that is a projecting wall feature item that strengthens a specific area of a casting. Each `Rib` is either a `General_rib` (see 4.2.135), or a `Planar_rib` (see 4.2.233).

EXAMPLE A rib is sometimes known as a bracket or a gusset to add stiffness between to flanges or between a hub and a flange.

### **4.2.272 Riser\_contact**

A `Riser_contact` is a type of `Gating_design_feature` (see 4.2.133) that is the shape and size of the feeder from the riser that attaches to the pattern. The data associated with `Riser_contact` are the following:

- `break_off_connection`;
- `contact_placement`;

#### **4.2.272.1 break\_off\_connection**

The `break_off_connection` is the size and shape of just the point of break off from the casting as it is attached to the pattern. See 4.3.499 for the application assertion.



### 4.2.272.2 contact\_placement

The `contact_placement` specifies that the riser contact is located on the cope side or the drag side or both.

NOTE the riser will always be on the cope side but the riser basin or well will always be on the drag side.

NOTE See 4.2.272.2.1 to 4.2.272.2.3 for the definition of each allowable value for `contact_placement` type.

**4.2.272.2.1 cope:** the ingate is located in the cope side of the mould.

**4.2.272.2.2 drag:** the ingate is located in the drag side of the mould.

**4.2.272.2.3 cope\_and\_drag:** the ingate is located in both the cope and drag side of the mould.

### 4.2.273 Riser

A Riser is a type of `Gating_design_feature` (see 4.2.133) that is a reservoir of molten metal to be fed back into the casting as solidification occurs to prevent shrinkage.

NOTE 1 Figure 18 illustrates a Riser and associated `Gating_system`.

NOTE 2 This application object is harmonized with the riser object from ISO 8062-1:2007, Clause 3.6.

The data associated with Risers are the following:

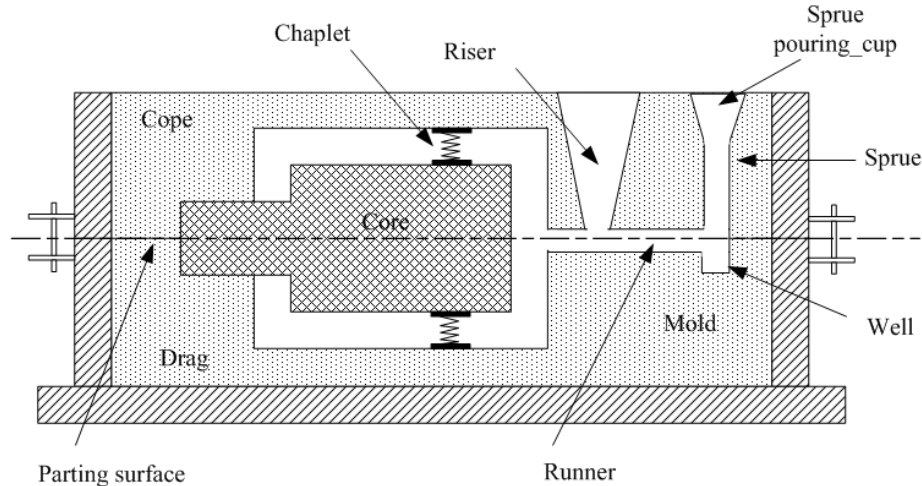
- `contact_definition`;
- `riser_dimensions`;
- `riser_core`;
- `riser_neck`;
- `riser_type`;
- `riser_sleeve`;
- `riser_well`.

#### 4.2.273.1 contact\_definition

The `contact_definition` defines the type and form of the feeder from the riser to the contact area at the casting or as attached to the production pattern. See 4.3.501 for the application assertion.

### 4.2.273.2 riser\_core

The `riser_core` specifies the core size that is used in mould with a riser and provide a break-off for the riser from the casting. The `riser_core` need not be specified for a particular Risers. See 4.3.500 for the application assertion.



**Figure 18 — Riser connection to Gating\_system**

### 4.2.273.3 riser\_dimensions

The `riser_dimensions` specifies the size and form with dimensional values for the riser. See 4.3.500 for the application assertion.

### 4.2.273.4 riser\_neck

The `riser_neck` specifies the neck size that connects to a riser and provides molten metal flow from the riser to a break-off contact for the Riser. There may be more than one `riser_neck` for a Risers. The `riser_neck` need not be specified for a particular Risers. See 4.3.500 for the application assertion.

#### 4.2.273.5 riser\_sleeve

The riser\_sleeve specifies the kind of tube enclosure used in the mould with a riser to assist the metal to stay liquid in the riser during back-feed to the casting. The riser\_sleeve need not be specified for a particular Riser.

NOTE See 4.2.273.5.1 to 4.2.273.5.3 for the definition of each allowable value for type\_used.

**4.2.273.5.1 insulating:** a sleeve that serves as an insulator to reduce the rate of heat loss from the sides of the riser.

**4.2.273.5.2 exothermic:** a sleeve that generates some heat from a chemical material within the sleeve that reacts to the molten metal.

**4.2.273.5.3 exothermic\_insulating:** a sleeve made of exothermic materials surrounded by an insulating material.

#### 4.2.273.6 riser\_type

The riser\_type specifies the kind or style of riser to be used for the cast part.

EXAMPLE open, closed, thermal are some of the riser types.

NOTE See 4.2.273.6.1 to 4.2.273.6.5 for the definition of each allowable value for riser\_type.

**4.2.273.6.1 blind:** a riser that is totally enclosed inside the mould.

**4.2.273.6.2 open:** a side riser that is open to the top of the mould and to the atmosphere.

**4.2.273.6.3 open\_with\_radiation\_sheld:** a open side riser that has an insulating or exothermic material placed on top of the mould to cover the riser.

**4.2.273.6.4 top:** a riser that is located on top of the casting.

**4.2.273.6.5 top\_with\_radiation\_sheld:** a top riser that has an insulating or exothermic material placed on top of the mould to cover the riser.

#### 4.2.273.7 riser\_well

The riser\_well specifies the depth of the well on the drag side of the pattern plate. See 4.3.502 for the application assertion.

NOTE the well for the riser is located directly under the riser but normally slightly larger than the riser at the cope parting surface.

### **4.2.274 Runner**

A Runner is a type of Gating\_design\_feature (see 4.2.133) that is a channel that connects the sprue to each ingate through which the molten metal flows into the mould cavity.

NOTE 1 The runner is sometimes referred to as a runner bar.

NOTE 2 This application object is harmonized with the runner object from ISO 8062-1:2007, Clause 3.3.

The data associated with Runner are the following:

- metal\_flow\_filter;
- runner\_choke;
- runner\_connection;
- runner\_dimensions;
- runner\_placement;
- runner\_vent;
- sprue\_connections.

#### **4.2.274.1 metal\_flow\_filter**

The metal\_flow\_filter specifies the type and size of Filter (see 4.2.126) used to strain or screen the molten metal before it enters the casting. The metal\_flow\_filter need not be specified for a Runner. See 4.3.505 for the application assertion.

#### **4.2.274.2 runner\_chock**

The runner\_chock specifies the size and location of the Choke (see 4.2.34) in the Runner. The runner\_chock need not be specified for a Runner. See 4.3.503 for the application assertion.

#### **4.2.274.3 runner\_connection**

The riser\_connection specifies the form and shape of the feeder from the runner to the riser. See 4.3.504 for the application assertion.

NOTE the feeder in this application is sometime called a gate from the runner to the riser that then makes contact with the production pattern.

#### 4.2.274.4 runner\_dimensions

The `runner_dimensions` specifies the configuration, form, and dimensional values for the runner. See 4.3.506 for the application assertion.

#### 4.2.274.5 runner\_placement

The `runner_placement` specifies location of the runner as being on the cope side of the `Pattern_plate` or the drag side or both.

NOTE when the runner overlap in the cope and drag it is intended that the amount will be passed the `runner_dimension` attribute.

NOTE See 4.2.274.5.1 to 4.2.274.5.3 for the definition of each allowable value for `contact_placement`.

**4.2.274.5.1 cope:** the ingate is located in the cope side of the mould.

**4.2.274.5.2 drag:** the ingate is located in the drag side of the mould.

**4.2.274.5.3 cope\_and\_drag:** the ingate is located in both the cope and drag side of the mould.

#### 4.2.274.6 runner\_vent

The `runner_vent` specifies size and location of the Vent (see 4.2.336) for a Runner. There may be more than one `runner_vent` for a Runner. The `runner_vent` need not be specified for a particular Runner. See 4.3.508 for the application assertion.

#### 4.2.274.7 sprue\_connections

The `sprue_connection` specifies the configuration and form of the runner where it makes contact with the sprue. See 4.3.507 for the application assertion.

NOTE the transition fillets are left to the pattern makers discretion.

#### 4.2.275 Sampled\_set

A `Sampled_set` is a group of parts that are inspected or tested together.

EXAMPLE In a particular process, every tenth casting is tested or examined thoroughly for a hardness, the value of a critical dimension, and other specified properties. The results of the inspection are presented daily for statistical process control. In this case the sample size is 10 per cent of the day's production.

ISO 10303-223:2008 (E)

The data associated with a `Sampled_set` are the following:

- `quality_assurance_document`;
- `master_samples_inspected`;
- `sampled_from`;
- `size`;

#### **4.2.275.1 `quality_assurance_document`**

The `quality_assurance_document` specifies one or more warrants or inspection reports that describe the characteristics of quality for the part. See 4.3.509 for the application assertion.

#### **4.2.275.2 `master_samples_inspected`**

The `master_sample_inspected` specifies the item that has been inspected. The `master_sample_inspected` need not be specified for a particular `Sampled_set`. See 4.3.510 for the application assertion.

#### **4.2.275.3 `sampled_from`**

The `sampled_from` specifies the Heat (see 4.2.143) or Lot (see 4.2.172) from which the item has been selected. See 4.3.511 and 4.3.512 for the application assertion.

NOTE The sample set of `master_samples` (usually castings) has been produced by removing a certain number of `master_samples` from a production lot, or from a production run made with a specific heat.

#### **4.2.275.4 `size`**

The `size` specifies the number of items in the `Sampled_set`. The `size` need not be specified for a particular `Sampled_set`.

#### **4.2.276 `Sand_casting_design_feature`**

A `Sand_casting_design_feature` is a type of `Feature` (see 4.2.124) that is used to produce a cast part. Each `Sand_casting_design_feature` is either a `Flask` (see 4.2.131), `Chaplet` (see 4.2.30), `Chaplet_pad` (see 4.2.31), `Pin_center` (see 4.2.229), `Mould_locks` (see 0), `Loose_piece` (see 4.2.171), `Chill` (see 4.2.33), `Casting_insert` (see 4.2.24), `Core` (see 4.2.48), `Sand_mould` (see 4.2.278), `Core_print` (see 4.2.57), `Mould_box` (see 4.2.199) or `Pattern_plate` (see 4.2.221). The data associated with a `Sand_casting_design_feature` are the following:

- `placement`

### 4.2.276.1 placement

The placement specifies the position and orientation of a Sand\_casting\_design\_feature relative to the base shape for a part. See 4.3.513 for the application assertion.

### 4.2.277 Sand\_casting\_tooling

A Sand\_casting\_tooling is a type of Tooling (see 4.2.333) that defines the resources necessary to produce a cast part. The data associated with a Sand\_casting\_tooling are the following:

- pattern\_equipment;
- pattern\_number.

#### 4.2.277.1 pattern\_equipment

The pattern\_equipment specifies pattern equipment supplied by the customer necessary to manufacture a cast part. See 4.3.514 for the application assertion.

#### 4.2.277.2 pattern\_number

The pattern\_number specifies an alphanumeric string that identifies the pattern used to produce the cast part.

### 4.2.278 Sand\_mould

A Sand\_mould is a type of Sand\_casting\_design\_feature (see 4.2.276) that is a type of casting into which liquid metal is poured and allowed to solidify to form a cast part.

NOTE 1 Normally the Sand\_mould consists of two halves (the cope and drag portions of the mould), but in some cases there is more than one parting line and the Sand\_mould definition is more complex.

NOTE 2 Items such as cores, chills, chaplets, insulating sleeves, in-mould filters, or in mould additions of flux, grain refiner, or inoculant, may be placed in a Sand\_mould.

The data associated with a Sand\_mould are the following:

- chaplet\_definition;
- chill\_definitions;
- mould\_made\_by;
- mould\_defined\_for;
- sand\_mould\_type.

### 4.2.278.1 chaplet\_definition

The chaplet\_definition specifies the chaplet requirements to support or stabilize a core for the cast part. See 4.3.516 for the application assertion.

### 4.2.278.2 chill\_definition

The chill\_definition specifies the location for positioning the chill. See 4.3.517 for the application assertion.

NOTE the proper location of the chills for a cast part is important to ensure hardness or solidification control when required.

### 4.2.278.3 mould\_made\_by

The mould\_made\_by defines the equipment necessary to make the Sand\_mould. See 4.3.515 for the application assertion.

### 4.2.278.4 sand\_mould\_type

The sand\_mould\_type specifies the type of sand mould.

EXAMPLE the sand\_mould\_type could be green sand, no-bake, shell mould, flaskless are examples of the types of sand moulds.

NOTE See 4.2.278.4.1 to 4.2.278.4.5 for the definition of each allowable value for sand\_mould\_characteristics type.

**4.2.278.4.1 cold box mould method:** to produce a mould by the use of a vaporized catalyst to cure a resin-coated sand.

NOTE Often referred to as CO2 moulding process.

**4.2.278.4.2 green sand material:** to produce a mould by the use of a naturally bonded sand or a compounded moulding sand mixture that has been tempered with water.

**4.2.278.4.3 no bake mould method:** to produce a mould by the use of a chemically bonded sand that requires a liquid catalyst to cure the binder at ambient temperature over a specified period of time.

**4.2.278.4.4 skin dried method:** to produce a mould by the use of green sand moulding and the surface of the mould is heated to remove all moisture content from the sand mould surface.

NOTE this application is used primarily in large ferrous metalcasting applications.

**4.2.278.4.5 vacuum method:** to produce a mould using the V-process.



### 4.2.279 Secondary\_tooling

A Secondary\_tooling is a type of die\_design\_feature (see 4.2.83) that is a supplementary devices used to facilitate the casting process. The data associated with a Secondary\_tooling are the following:

— type\_of\_tooling

#### 4.2.279.1 type\_of\_tooling

The type\_of\_tooling specifies the kind of secondary tooling.

EXAMPLE "Extraction gripper", "trim press", and "grinding fixture" are kinds of Secondary\_tooling.

### 4.2.280 Setup\_activity

An Activity\_setup is a type of Activity (see 4.2.1) that defines preparing the equipment, or other casting devices for a casting process. Each Activity\_setup is either an Ancillary\_setup (see 4.2.9) or Equipment\_setup (see 4.2.113).

NOTE This application object is harmonized with the entity setup\_activity from ISO 10303-240:2005, Clause 4.2.128.

### 4.2.281 Shakeout

The Shakeout is a type of Equipment (see 4.2.110) that is a mechanical method for removing castings from mould.

### 4.2.282 Shape\_aspect

A Shape\_aspect is a region of interest with respect to the shape of a part. A Shape\_aspect may be an element of the shape of the part or a reference shape that does not lie on the shape of the part, but is used to specify a characteristic of the shape of the part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.218.

The data associated with a Shape\_aspect are the following:

- element;
- representation\_form;
- representation\_shape.

#### 4.2.282.1 element

The element specifies Shape\_element (see 4.2.285) that are components of the shape of the Part. The element need not be specified for a particular Shape\_aspect. See 4.3.518 for the application assertion.

### **4.2.282.2 representation\_form**

The `representation_form` specifies aspects of the boundary representation of the Part. There may be more than one `representation_form` for a `Shape_aspect`. See 4.3.520 for the application assertion.

### **4.2.282.3 representation\_shape**

The `representation_shape` specifies the boundary representation of the shape of the Part. The `representation_shape` need not be specified for a particular `Shape_aspect`. There may be more than one `representation_shape` for a `Shape_aspect`. See 4.3.519 for the application assertion.

## **4.2.283 Shape\_aspect\_representation**

A `Shape_aspect_representation` is a grouping of geometric elements with respect to the representation of a part.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.130.

The data associated with a `Shape_aspect_representation` are the following:

— `shape_definition`.

### **4.2.283.1 shape\_definition**

The `shape_definition` specifies the `Geometric_model` (see 4.2.137) to define the shape that is the representation. See 4.3.521 for the application assertion.

## **4.2.284 Shape\_dimensions**

The `Shape_dimensions` are the modified shape dimensions that are required to produce a cast part and still satisfy the form, fit and function specified by the customer.

NOTE the modified dimensions are the result of required draft angles, parting line surfaces, metal shrinkage compensation values, machine stock allowances, and metal flow transitions within the cast part that alters the part shape as designed.

The data associated with a `Shape_dimensions` are the following:

— `dimensionals`;

— `tooling`.

### **4.2.284.1 dimensionals**

The `dimensionals` specifies the modified measurement values for the altered part shape.

NOTE See 4.2.284.1.1 to 4.2.284.1.5 for the definition of each allowable value for `dimensionals` types.

**4.2.284.1.1 added:** are dimensions that are added to the part design to permit castability

EXAMPLE: this could be dimensions for the gating system, for core print sizes, and others

**4.2.284.1.2 auxiliary:** Restrict auxiliary exact dimension. Dimension\_tolerance to be a nominal value with no value\_limitation.

**4.2.284.1.3 modified:** are dimensions that are modified for draft, machine allowances, and shrink factors.

**4.2.284.1.4 theoretical:** Restrict theoretically exact dimension. Dimension\_tolerance to be a nominal value with no value\_limitation.

**4.2.284.1.5 user defined:** a description specified by the user.

## 4.2.284.2 tooling

The tooling specifies the specific the required tool for the moulding process to produce the cast part. See 4.3.522, 4.3.523, 4.3.526 for the application assertion.

## 4.2.285 Shape\_element

A Shape\_element is a specific kind of Shape\_aspect that identifies a portion of a shape and its representation. Each Shape\_element is either a Direction\_element (see 4.2.92), Drafted\_surface (see 4.2.98), Edge\_shape\_surface (see 4.2.100), Face\_shape\_element (see 4.2.122), Location\_element (see 4.2.168), Feature (see 4.2.124), Parting\_surface (see 4.2.218), Path\_element (see 4.2.219) or a Planar\_element (see 4.2.232).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.219.

## 4.2.286 Shape\_representation\_type

A Shape\_representation\_type is the types of geometric representations that may be used to define product shape.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.131.

The data associated with a Shape\_representation\_type are the following:

— geometry\_type.

### **4.2.286.1 geometry\_type**

The `geometry_type` specifies the types of geometric representation. The value of the `geometry_type` shall be one of the following:

— `advanced_boundary_rep`.

NOTE See 4.2.286.1.1 for the definition of each allowable value for `geometry_type`.

**4.2.286.1.1 advanced\_boundary\_rep:** the geometric representation of the shape, or an aspect of the shape, by an advanced boundary representation solid model. The geometric representation allows for the definition of curves and surfaces and for the topology that bounds them. Boundaries are explicitly defined only by topology. All of the geometry that defines the shape or shape aspect of the object shall be associated with topology.

### **4.2.287 Shop\_work\_order**

A `Shop_work_order` is a document that contains the information to process a part on the shop floor, and track the part's progress throughout its manufacture.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.221.

The data associated with a `Shop_work_order` are the following:

— `order_id`.

#### **4.2.287.1 order\_id**

The `order_id` specifies a unique identifier for the `Shop_work_order`.

### **4.2.288 Shot\_sleeve**

The `Shot_sleeve` is a type of `die_design_feature` (see 4.2.83) that is the conduit from the diecasting machine that provides molten metal to die mould cavity. The data associated with a `Shot_sleeve` are the following:

— `slot_sleeve_size`.

#### **4.2.288.1 slot\_sleeve\_size**

The `slot_sleeve_size` specifies the dimensional parameters for the sleeve that provides molten metal to the production mould. See 4.3.527 for the application assertion.

### **4.2.289 Simulated\_property**

A `Simulated_property` is a physical variable whose value is predicted by a simulation.

EXAMPLE Temperature, fraction liquid, velocity, and state of stress.

The data associated with a `Simulated_property` are the following:

- `name`;
- `property_value`.

#### **4.2.289.1 name**

The name specifies the label used to identify the `Simulated_property`.

#### **4.2.289.2 property\_value**

The `property_value` specifies the numerical quantity that characterizes the `Simulated_property`.

NOTE See 4.2.289.2.1 to 4.2.289.2.3 for the definition of each allowable value for `value_select`.

**4.2.289.2.1 Scalar\_measure:** is a value that can be characterized by a single numerical quantity.

**4.2.289.2.2 Tensor\_measure:** is a quantitative measure specifying position within more than one coordinate system.

**4.2.289.2.3 Vector\_measure:** is the value of a variable that is characterized by a magnitude and a direction.

### **4.2.290 Simulation\_exception\_report**

A `Simulation_exception_report` is a type of `Document_assignment` (see 4.2.94) that outlines flaws or weaknesses in the part design or gating system that will cause imperfection in the cast part.

### **4.2.291 Simulation\_input**

A `Simulation_input` is the collection of data required by simulation software in order to begin a `Simulation_run` (see 4.2.298). The data associated with an assumption are the following:

- `assumption`;
- `condition`;
- `input_parameter`;

ISO 10303-223:2008 (E)

- integration\_time\_specification;
- region;
- setup\_file.

#### **4.2.291.1 assumption**

The assumption specifies the presumption about the state of physical phenomena for a Simulation\_input. The assumption need not be specified for a particular Simulation\_input. There may be more than one assumption for a Simulation\_input. See 4.3.528 for the application assertion.

#### **4.2.291.2 condition**

The condition specifies the constraints on physical phenomena for a Simulation\_input. The condition need not be specified for a particular Simulation\_input. There may be more than one condition for a Simulation\_input. See 4.3.528 for the application assertion.

#### **4.2.291.3 input\_parameter**

The input\_parameter specifies the values for a Simulation\_input. There may be more than one input\_parameter for a Simulation\_input. See 4.3.532 and 4.3.531 for the application assertion.

#### **4.2.291.4 integration\_time\_specification**

The integration\_time\_specification specifies the time factor need for a Simulation\_input. There may be more than one integration\_time\_specification for a Simulation\_input. See 4.3.529 for the application assertion.

#### **4.2.291.5 region**

The region specifies the particular section of the Simulation\_input. There may be more than one region for a Simulation\_input. See 4.3.533 for the application assertion.

#### **4.2.291.6 setup\_file**

The setup\_file specifies the identifies a document that holds the Simulation\_input. The setup\_file need not be specified for a particular Simulation\_input. See 4.3.530 for the application assertion.

## 4.2.292 Simulation\_input\_region

A Simulation\_input\_region is a portion of space containing similar material properties at the beginning of a Simulation\_run (see 4.2.298). The data associated with a Simulation\_input\_region are the following:

- associated\_part;
- boundary\_condition;
- meshing\_condition;
- region\_property;
- region\_type;
- simulation\_region\_shape;

### 4.2.292.1 associated\_part

The associated\_part specifies the Part\_version (see 4.2.217) objects which make up the Simulation\_input\_region. The associated\_part need not be specified for a particular Simulation\_input\_region. There may be more than one associated\_part for a Simulation\_input\_region. See 4.3.537 for the application assertion.

### 4.2.292.2 boundary\_condition

The boundary\_condition specifies the constraints on physical phenomena at the border of the Simulation\_input\_region. The boundary\_condition need not be specified for a particular Simulation\_input\_region. There may be more than one boundary\_condition for a Simulation\_input\_region. See 4.3.534 for the application assertion.

### 4.2.292.3 meshing\_condition

The meshing\_condition specifies the rules for dividing the Simulation\_input\_region into a mesh for simulation. The meshing\_condition need not be specified for a particular Simulation\_input\_region. There may be more than one meshing\_condition for a Simulation\_input\_region. See 4.3.535 for the application assertion.

### 4.2.292.4 region\_property

The region\_property specifies the value of a physical variable or a relationship between physical variables for the Simulation\_input\_region. There may be more than one region\_property for a Simulation\_input\_region. See 4.3.536 and 4.3.538 for the application assertion.

#### **4.2.292.5 region\_type**

The `region_type` specifies the kind of `Simulation_input_region`.

EXAMPLE "Empty cavity," "liquid metal," "cores," "mould," "mould coating," and "filter" are region types.

#### **4.2.292.6 simulation\_region\_shape**

The `simulation_region_shape` specifies the spatial form of the `Simulation_input_region`. See 4.3.539 for the application assertion.

### **4.2.293 Simulation\_output**

A `Simulation_output` is the data produced by a simulation software as the result of a `Simulation_run` (see 4.2.298). The data associated with a `Simulation_output` are the following:

- `predicted_defect`;
- `result`.

#### **4.2.293.1 predicted\_defect**

The `predicted_defect` specifies the an undesirable feature that the `Simulation_run` (see 4.2.298) indicates will occur. The predicted defect need not be specified for a particular `Simulation_output`. There may be more than one `predicted_defect` for a `Simulation_output`. See 4.3.540 for the application assertion.

#### **4.2.293.2 result**

The `result` specifies the a specific outcome of a `Simulation_run` (see 4.2.298). The `result` need not be specified for a particular `Simulation_output`. There may be more than one `result` for a `Simulation_output`. See 4.3.541 for the application assertion.

#### **4.2.294 Simulation\_output\_region**

A `Simulation_output_region` is a portion of space containing similar material properties at the end of a `Simulation_run` (see 4.2.298). The data associated with a `Simulation_output_region` are the following:

- `belongs_to`;
- `history`;
- `region_shape`.



#### 4.2.294.1 belongs\_to

The belongs\_to specifies the Simulation\_input\_region (see 4.2.292) of which the Simulation\_output\_region is a sub region. See 4.3.543 for the application assertion.

#### 4.2.294.2 history

The history specifies the predicted property values at different times during the simulation. See 4.3.544 for the application assertion.

#### 4.2.294.3 region\_shape

The region\_shape specifies the spatial form of the Simulation\_output\_region. See 4.3.542 for the application assertion.

### 4.2.295 Simulation\_process

A Simulation\_process is a type of Casting\_activity (see 4.2.21) that is the metalcasting process used by the foundry to be simulated by the simulation run. A Simulation\_process shall be either a Customer\_simulation (see 4.2.65) or a Metal\_caster\_simulation (see 4.2.196).

The data associated with a Simulation\_process are the following:

- part\_geometry;
- quality\_requirements;
- run\_simulation.
- tooling\_for\_simulation.

#### 4.2.295.1 part\_geometry

The part\_geometry specifies the design form of the Part\_version (see 4.2.217) for simulation to process. See 4.3.545, 4.3.546 and 4.3.547 for the application assertion.

#### 4.2.295.2 quality\_requirements

The quality\_requirements specifies the requirement set by the customer. There may be more than quality\_requirements for a Simulation\_process set by a Customer\_casting\_requirement (see 4.2.62). See 4.3.548 for the application assertion.

#### 4.2.295.3 run\_simulation

The run\_simulation specifies the input and output requirements for a Simulation\_run (see 4.2.298). See 4.3.549 for the application assertion.

#### **4.2.295.4 tooling\_for\_simulation**

The `tooling_for_simulation` specifies the necessary Tooling (4.2.333) needed for or to be used with the `Simulation_process`. See 4.3.550 for the application assertion.

#### **4.2.296 Simulation\_report**

A `Simulation_report` is a type of `Document_assignment` (see 4.2.94) that describes the results of a simulation study or analysis for a cast part.

#### **4.2.297 Simulation\_result**

A `Simulation_result` is a specific outcome during a specific time period of a `Simulation_run` (see 4.2.298).

The data associated with a `Simulation_result` are the following:

- `begin_time`;
- `end_time`;
- `illustration`;
- `region`.

##### **4.2.297.1 begin\_time**

The `begin_time` specifies the first simulation time reported by the `Simulation_result`.

##### **4.2.297.2 end\_time**

The `end_time` specifies the last simulation time reported by the `Simulation_result`.

##### **4.2.297.3 illustration**

The `illustration` specifies the picture that contains the output of a `Simulation_result`. The `illustration` need not be specified for a particular `Simulation_result`. There may be more than one `illustration` for a `Simulation_result`. See 4.3.551 for the application assertion.

##### **4.2.297.4 region**

The `region` specifies the `Simulation_output_region` (see 4.2.294) in which the `Simulation_result` occurred. There may be more than one `region` for a `Simulation_result`. See 4.3.552 for the application assertion.

## 4.2.298 Simulation\_run

A Simulation\_run is an execution of a simulation software program to model a casting process. The data associated with a Simulation\_run are the following:

- associated\_process;
- description;
- identification;
- input;
- output;
- software\_used.

### 4.2.298.1 associated\_process

The associated process specifies the Activity (see 4.2.1) modeled by the Simulation\_run. The associated process need not be specified for a particular Simulation\_run. See 4.3.553 for the application assertion.

### 4.2.298.2 description

The description specifies the textual account of the Simulation\_run. There may be more than one description for a Simulation\_run.

### 4.2.298.3 identification

The identification specifies the label used to identify a Simulation\_run.

### 4.2.298.4 input

The input specifies the data used by the simulation software to begin a Simulation\_run. See 4.3.554 for the application assertion.

### 4.2.298.5 output

The output specifies the data produced by the simulation software as the result of a Simulation\_run. See 4.3.556 for the application assertion.

### 4.2.298.6 software\_used

The software used specifies the computer program used to model the casting process. See 4.3.556 for the application assertion.

## **4.2.299 Simulation\_run\_relationship**

A `Simulation_run_relationship` is an association between two different `Simulation_run` (see 4.2.298) objects. The data associated with a `Simulation_run_relationship` are the following:

NOTE The meaning of the relationship is given by the description attribute.

- `description`;
- `related`;
- `relating`.

### **4.2.299.1 description**

The `description` specifies the textual account of the `Simulation_run_relationship`.

### **4.2.299.2 related**

The `related` specifies the first `Simulation_run` (see 4.2.298) in the relationship. See 4.3.557 for the application assertion.

### **4.2.299.3 relating**

The `relating` specifies the second `Simulation_run` (see 4.2.298) in the relationship. See 4.3.557 for the application assertion.

## **4.2.300 Simulation\_software**

A `Simulation_software` is the computer program used to model the casting process.

NOTE The `Simulation_software` mimics the behavior of the casting process and shows values of various process variables at various times.

The data associated with a `Simulation_software` are the following:

- `release_date`;
- `software_name`;
- `software_version`.

**4.2.300.1 release\_date**

The `release_date` specifies the date on which the Simulation software was made available for use. The `release_date` need not be specified for a particular Simulation\_software. See 4.3.567 for the application assertion.

**4.2.300.2 software\_name**

The `software_name` specifies the label used to identify the Simulation\_software.

**4.2.300.3 software\_version**

The `software_version` specifies the variant of the Simulation\_software used for the simulation. The `software_version` need not be specified for a particular Simulation\_software.

**4.2.301 Simulation\_tooling\_report**

A Simulation\_tooling\_report is a type of Document\_assignment (see 4.2.94) that outlines changes that could be made to the tooling for the part design or gating system that would improve production rates and produce a quality cast part.

**4.2.302 Simulation\_unit**

A Simulation\_unit is a point or region which is not broken down further for the purpose of simulation. The data associated with a Simulation\_unit are the following:

- `belongs_to`;
- `history`;
- `name`;
- `unit_shape`;
- `unit_location`.

**4.2.302.1 belongs\_to**

The `belongs_to` specifies the Simulation\_output\_region (see 4.2.294) containing the Simulation\_unit. See 4.3.560 for the application assertion.

**4.2.302.2 history**

The `history` specifies the value of a simulated property as a function of simulation time. There may be more than one history for a Simulation\_unit. See 4.3.561 for the application assertion.

### **4.2.302.3 name**

The name specifies the label used to identify the Simulation\_unit.

### **4.2.302.4 unit\_location**

The unit\_location specifies the position of the Simulation\_unit. See 4.3.559 for the application assertion.

### **4.2.302.5 unit\_shape**

The unit\_shape specifies the spatial form occupied by the Simulation\_unit. The unit\_shape need not be specified for a particular Simulation\_unit. See 4.3.562 for the application assertion.

## **4.2.303 Simulation\_unit\_state**

A Simulation\_unit\_state is the condition of the material inside a Simulation\_unit (see 4.2.302) at a particular value of simulation time. The data associated with a Simulation\_unit\_state are the following:

- evaluation\_time;
- property.

### **4.2.303.1 evaluation\_time**

The evaluation\_time specifies the simulation time at which property values were sampled.

### **4.2.303.2 property**

The property specifies the physical value that was computed at the evaluation time for a Simulation\_unit (see 4.2.302). There may be more than one property for a Simulation\_unit\_state. See 4.3.563 for the application assertion.

## **4.2.304 Size\_tolerance**

A Size\_tolerance is a type of Dimensional\_tolerance (see 4.2.91) that is the size dimension tolerance characteristic for a geometric element. Each Size\_tolerance is either an Angular\_size\_dimension\_tolerance (see 4.2.11), Curved\_dimension\_tolerance (see 4.2.61), Diameter\_dimension\_tolerance (see 4.2.78), Height\_dimension (see 4.2.146), Length\_dimension (see 4.2.164), Width\_dimension (see 4.2.339), Radial\_dimension\_tolerance (see 4.2.260) Thickness\_tolerance (see 4.2.323), or an Externally\_defined\_size\_dimension (see 4.2.120).

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.223.

The data associated with a `Size_tolerance` are the following:

- `applied_shape`;
- `envelope`.

#### **4.2.304.1 applied\_shape**

The `applied_shape` specifies the physical shape of the Part that is being toleranced. See 4.3.564 for the application assertion.

#### **4.2.304.2 envelope**

The `envelope` specifies that each geometric constraint has to be fulfilled in itself. The envelope of the perfect shape corresponding to the maximum material shall not be larger than the specified dimension and tolerance. The `envelope` attribute shall be a boolean value, if TRUE the envelope is required for the `Tolerance_value`.

#### **4.2.305 Slide**

A `Slide` is a type of `die_design_feature` (see 4.2.83) that is a metal object that is used in a die mould to form some portion of a cast part that can not be form and drawn by the die mould.

NOTE This application object is harmonized with the `slide` object from ISO 8062-1:2007, Clause 2.6.

The data associated with a `Slide` are the following:

- `slide_activation_method`;
- `slide_material`;
- `shape`.

##### **4.2.305.1 slide\_activation\_method**

The `slide_activation_method` specifies and describes in human interruptible text the method used to activate the slide mechanism.

##### **4.2.305.2 slide\_material**

The `slide_material` specifies the substance used to manufacture the slide.

NOTE See 4.2.183.2.1 to 4.2.183.2.16 for the definition of each allowable value for material type

### 4.2.305.3 shape

The shape specifies one or more geometrical shapes and forms with the dimensional values of the slide. See 4.3.565 or the application assertion.

### 4.2.306 Special\_inspection\_requirement

A Special\_inspection\_requirement is a type of Customer\_casting\_requirement (see 4.2.62) that is a specific method for evaluation and qualification of the cast part for acceptance by the customer. The data associated with a Special\_inspection\_requirement are the following:

- inspection\_requirements;
- inspection\_description;
- other\_testing\_method;
- performed\_by\_customer.

#### 4.2.306.1 inspection\_requirements

The inspection\_requirement specifies one or more specifications or standards that define how the inspection or testing procedure is to be applied. See 4.3.566 for the application assertion.

#### 4.2.306.2 inspection\_description

The inspection\_description specifies the special inspection method to be used to inspect the cast part for soundness of metal.

NOTE See 4.2.306.2.1 to 4.2.306.2.6 for the definition of each allowable value for inspection\_description type.

**4.2.306.2.1 liquid penetrant:** a testing method that uses dyes that penetrate the casting to detect minute imperfections not visible during visual inspection.

**4.2.306.2.2 magnetic particle:** an inspection method that can only be used with alloys that can be magnetized to detect small cracks on or near the surface of a casting.

**4.2.306.2.3 pressure test:** an inspection method that uses either pressurized air or liquid to detect voids that would cause the cast part to leak, sometimes called a leak test.

**4.2.306.2.4 radiography:** an x-ray inspection method for detecting internal shrinkage, porosity, dross, or slag film not detectable by visual inspection methods.



**4.2.306.2.5 testing method other:** specific customer defined inspection or testing methods not in this list.

**4.2.306.2.6 ultrasonic:** an inspection method that uses high-frequency acoustic energy with transducers to identify or locate internal defects in a cast part.

### **4.2.306.3 performed\_by\_customer**

The `performed_by_customer` specifies one or more boolean values that indicate if any of the special inspection is performed by the customer. A value of true specifies that the inspection is performed by the customer. A value of false specifies the required inspection will be performed by the foundry or their inspection agent.

### **4.2.306.4 other\_testing\_method**

The `other_testing_method` states what specific testing method is required that is not identified by the inspection types. The `other_testing_method` need not be specified for a particular `Special_inspection_requirement` for the cast part.

NOTE If the attribute `inspection_description` contains a value `testing_method_other` for the entity `Special_inspection_requirement` then the attribute `other_testing_method` is not optional.

## **4.2.307 Special\_instruction**

A `Special_instruction` is a description of some aspect of either an Activity (see 4.2.1) that requires further explanation be provided to the individual performing the corresponding Activity.

NOTE 1 This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.135.

NOTE 2 This is different from the Activity description which describes the general instructions or objectives of the activity. This is also distinctive from an external procedure which gives detailed steps necessary to perform some procedure.

The data associated with a `Special_instruction` is the following:

- `instruction_text`;
- `instruction_type`.

### **4.2.307.1 instruction\_text**

The `instruction_text` specifies the word, or group of words, that applies to an activity describing the procedural aspect of the special instructions or conditions which shall be met or observed.

### **4.2.307.2 instruction\_type**

The `instruction_type` specifies one word or group of words that categorize the instruction.

EXAMPLE The included types of categories are warning, hazardous and environmental.

### **4.2.308 Specification**

A Specification is a document that defines information pertaining to properties or processes for a part or an aspect of a part. A Specification may be a `Supplemental_document` (see 4.2.314)

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.226.

The data associated with a Specification are the following:

- `constraint`;
- `issuing_organization`;
- `release_date`;
- `specification_class`;
- `specification_description`;
- `specification_document`;
- `specification_id`;
- `version_id`;

#### **4.2.308.1 constraint**

The `constraint` specifies the restriction on the Specification. The `constraint` need not be specified for a particular Specification. There may be more than one `constraint` for a Specification. See 4.3.568 for the application assertion.

#### **4.2.308.2 release\_date**

The `release_date` specifies the date of the applicable version of the specification. The `release_date` need not be specified. See 4.3.569 for the application assertion.

### 4.2.308.3 **issuing\_organization**

The `issuing_organization` specifies the organization that published the specification. See 4.3.569 for the application assertion.

EXAMPLE Examples of issuing organizations are ISO, ANSI, DIN.

### 4.2.308.4 **specification\_class**

The `specification_class` specifies a section within a Specification that is divided into classes. A Specification may but need not require a `specification_class`.

### 4.2.308.5 **specification\_description**

The `specification_description` specifies in human interpretable prose a description of the contents of the specification and any notes with respect to the Specification. A Specification may but need not require a `specification_description`.

### 4.2.308.6 **specification\_document**

The `specification_document` specifies the Document\_assignment (see 4.2.94) that is used to provide information. A `specification_document` need not be specified for a particular Specification. See 4.3.570 for the application assertion.

### 4.2.308.7 **specification\_id**

The `specification_id` specifies a unique identifier of the document.

### 4.2.308.8 **version\_id**

The `version_id` specifies the revision or issue number that uniquely distinguishes one Specification object from other versions of Specification objects with the same `specification_id`. The `version_id` need not be specified.

### 4.2.309 **Specification\_usage\_constraint**

A `Specification_usage_constraint` is a restriction on the application of information defined within a Specification.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.227.

ISO 10303-223:2008 (E)

The data associated with a `Specification_usage_constraint` are the following:

- `class_id`;
- `element`.

#### **4.2.309.1 class\_id**

The `class_id` specifies the data or range of data with respect to the element that defines the restriction imposed on the usage of the Specification.

#### **4.2.309.2 element**

The `element` specifies the particular piece or area of information that is being restricted within the Specification.

#### **4.2.310 Sprue**

The `Sprue` is a type of `Gating_design_feature` (see 4.2.133) that is an opening from the top of the mould to the gating system and is where the molten metal enters the mould cavity.

NOTE 1 Figure 19 illustrates a `Sprue` and associated `Gating_system`.

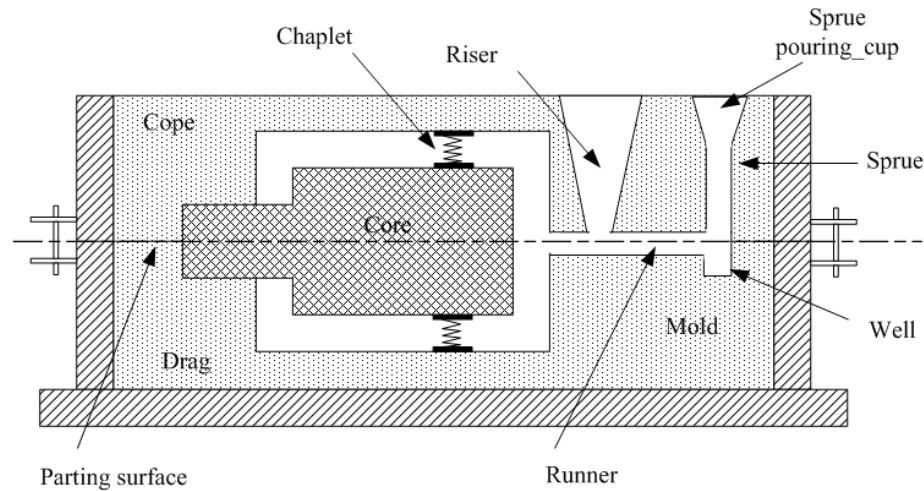
NOTE 2 This application object is harmonized with the `sprue` object from ISO 8062-1:2007, Clause 3.1.

The data associated with a `Sprue` are the following:

- `metal_flow_filter`;
- `pouring_cup`;
- `sprue_core`;
- `sprue_dimensions`;
- `sprue_well`.

##### **4.2.310.1 metal\_flow\_filter**

The `metal_flow_filter` specifies the type of filter used to strain the molten metal before it enters the casting. See 4.3.572 for the application assertion.



**Figure 19 — Sprue connection to Gating system**

#### 4.2.310.2 pouring\_cup

The pouring\_cup specifies a cup-shaped depression in a mould that is placed over a sprue opening and acts as a funnel to receive the molten metal from the ladle. The pouring\_cup need not be specified for a particular Sprue. There may be more than one pouring\_cup for a Sprue. See 4.3.571 for the application assertion.

#### 4.2.310.3 sprue\_core

The sprue\_core specifies a choke core at the bottom of the sprue opening that controls the flow rate of metal from the sprue. The sprue\_core need not be specified for a particular Sprue.

#### 4.2.310.4 sprue\_dimensions

The sprue\_dimensions specifies the dimensional values for the sprue. See 4.3.572 for the application assertion.

#### 4.2.310.5 sprue\_well

The sprue\_well specifies the depth of the well on the drag side of the pattern plate. See 4.3.573 for the application assertion.

NOTE the well for the sprue is located directly under the sprue button but normally slightly larger than the sprue at the cope parting surface.

### **4.2.311 Sprue\_and\_runner\_mould**

A `Sprue_and_runner_mould` is a type of `investment_design_feature` (see 4.2.161) that is the die that forms the sprue and the runner in wax or EPS to which the ingate and pattern(s) is attached. The data associated with a `Sprue_and_runner_mould` are the following:

- `gate_runner`;
- `mould_assembly`;
- `pouring_sprue`;
- `rigging_component`.

#### **4.2.311.1 gate\_runner**

The `gate_runner` specifies a rod that supports the model while the mould is being made. See 4.3.576 for the application assertion.

#### **4.2.311.2 mould\_assembly**

The `mould_assembly` specifies the `Investment_mould` (see 4.2.162) structure and arrangement. See 4.3.575 for the application assertion.

#### **4.2.311.3 pouring\_sprue**

The `pouring_sprues` provide a passageway for melted wax to leave the mould and also provide the entrance through which molten metal is poured into the mould. See 4.3.577 for the application assertion.

#### **4.2.311.4 rigging\_component**

The `rigging_component` specifies the `Ingate` (see 4.2.152) items connecting the pattern to the `Sprue` and `runner`. See 4.3.574 for the application assertion.

### **4.2.312 Status\_authority**

A `Status_authority` is the identification of completeness and availability of the item for use.

NOTE `Status_authority` for a given `Activity` can accommodate the requirement to convey information such as: Originator - Jane Doe, April 1, 1992; Checked by - Ted E. Bear, May 4, 1991. This requires that there can be many `Status_authorities` for one `Activity`.

The data associated with a Status\_ authority are the following:

- approval\_title;
- date;
- is\_given\_by.

#### **4.2.312.1 approval\_title**

The approval\_title specifies the item within a category of potential approvals that applies for this Status\_ authority. The category consists of process planner, checker, or other Process\_plan\_version release authorities.

#### **4.2.312.2 date**

The date specifies the calendar date on which the approval\_title was given.

#### **4.2.312.3 is\_given\_by**

The is\_given\_by specifies the Person\_in\_organization (see 4.2.228) who has the authority to approve process plans. See 4.3.578 for the application assertion.

### **4.2.313 Stop\_pin**

A Stop\_pin is a type of Ejector\_design\_feature (see 4.2.102) that is a pin(s) attached to the Ejector\_plate that limits the retraction travel distance and may also limit the ejection travel distance. The data associated with a Stop\_pin are the following:

- size.

#### **4.2.313.1 size**

The size specifies the dimensional information for the pin size used in the ejector system. See 4.3.579 for the application assertion.

### **4.2.314 Supplemental\_document**

A Supplemental\_document is a type of Specification (see 4.2.308) that is a document describing a specific methodology, policy or procedure. These documents, unique to each company, establish engineering and technical limitations and applications for design and engineering, materials, processes, and methods.

NOTE 1 This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.138.

NOTE 2 These documents make up the policy statements and procedures that are followed by the organization. These policies and procedures serve as guides to the employees of the organization and specify actions to be taken under certain conditions.

© ISO 2008 — All rights reserved

279

ISO 10303-223:2008 (E)

These procedures are considered to be external to the process planning function, and are directly related to business practice in the enterprise.

#### **4.2.315 Supplier\_BSU**

A `Supplier_BSU` (supplier basic semantical unit) is a type of `BSU` (see 4.2.18) that identifies the supplier of a parts library.

#### **4.2.316 Surface\_finish**

In `Surface_finish` is a type of `Non_casting_operations` (see 4.2.202) that identifies the activity of applying a finish to the cast part.

#### **4.2.317 Surface\_property**

The `Surface_property` specifies characteristics of a surface that are elements of the shape of a part.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.238.

The data associated with a `Surface_property` are the following:

- `casting_appearance`;
- `property_characteristics`;
- `surface_finish`.

##### **4.2.317.1 casting\_appearance**

The `casting_appearance` specifies the value placed on how the casting looks visually.

NOTE See 4.2.317.1.1 to 4.2.317.1.3 for the definition of each allowable value for casting appearance type.

**4.2.317.1.1 critical:** the casting appearance is blemish free with very little grinding at parting lines, core lines, or flashing.

**4.2.317.1.2 important:** the casting has minor defects and blemishes minor grinding permitted.

**4.2.317.1.3 not\_important:** casting appearance is not of value. The casting in use is not visible.

##### **4.2.317.2 property\_characteristics**

The `property_characteristics` specifies the parameter to describe the `Surface_property`. See 4.3.580 for the application assertion.



### 4.2.317.3 surface\_finish

The `surface_finish` specifies a boolean value that indicates a type of `Surface_property` is a surface finish. A value of true specifies the `Surface_property` is a surface finish property, a value of false specifies the `Surface_property` is for other surface properties.

### 4.2.318 Target\_area

A `Target_area` is a type of `Datum_target` (see 4.2.71) that is an enclosed area bounded by an arbitrary shape required to define a `Datum_target`. The shape of the `Target_area` is described explicitly by a set of curves.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.240.

The data associated with `Target_area` are the following:

— `area_shape`.

#### 4.2.318.1 area\_shape

The `area_shape` specifies the physical form of the `Target_area` shape. See 4.3.581 for the application assertion.

### 4.2.319 Target\_circle

A `Target_circle` is a type of `Placed_target` (see 4.2.231) that is an enclosed area bounded by a circle required to define a `Datum_target`. The origin of the `Datum_target` is the center of the circle, and the orientation is the x-y plane.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.241.

The data associated with a `Target_circle` are the following:

— `target_diameter`.

#### 4.2.319.1 target\_diameter

The `target_diameter` specifies the diameter value of the `Target_circle`.

### 4.2.320 Target\_line

A `Target_line` is a type of `Placed_target` (see 4.2.231) that is a straight curve. The origin shall be the first end point of the `Target_line`, the second end point shall be located on the Z-axis at a specified length.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.242.

ISO 10303-223:2008 (E)

The data associated with a Target\_line are the following:

— target\_length.

#### **4.2.320.1 target\_length**

The target\_length specifies the length value of the Target\_line.

#### **4.2.321 Target\_point**

A Target\_point is a type of Placed\_target (see 4.2.231) that is a single point. The origin shall be at the Target\_point.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.243.

#### **4.2.322 Target\_rectangle**

A Target\_rectangle is a type of Placed\_target (see 4.2.231) that is an area bounded by four sides with opposite sides equal in length. The center of the rectangle is at the origin. The orientation of the rectangle is with the length along the X-axis and the width along the Y-axis.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.244.

The data associated with Target\_rectangle are the following:

— target\_length;

— target\_width.

##### **4.2.322.1 target\_length**

The target\_length specifies the length value of the Target\_rectangle.

##### **4.2.322.2 target\_width**

The target\_width specifies the width value of the Target\_rectangle.

#### **4.2.323 Thickness\_tolerance**

A Thickness\_tolerance is a type of Size\_tolerance (see 4.2.304) that represents a thickness.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.248.

The data associated with a Thickness\_tolerance are the following:

— path.

### 4.2.323.1 path

The path specifies the shape that the tolerance applies to. A Thickness\_tolerance may but need not require a path. See 4.3.582 for the application assertion.

### 4.2.324 Tolerance\_inspection

The Tolerance\_inspection is a type of Inspection\_or\_test\_result (see 4.2.155) that is the outcome of a measurement and comparing the resultant tolerance to the design tolerance requirement.

The data associated with a Tolerance\_inspection are the following:

- requirement;
- tolerance\_result.

#### 4.2.324.1 requirement

The requirement specifies the Tolerance\_requirement (see 4.2.328) that was evaluated to produce the result. See 4.3.583 for the application assertion.

#### 4.2.324.2 tolerance\_result

The tolerance\_result specifies the outcome of the Tolerance inspection against the requirements. See 4.3.584 for the application assertion.

### 4.2.325 Tolerance\_limit

A Tolerance\_limit is an upper or lower tolerance value applied directly to a dimension. When applied to a Dimensional\_tolerance, the dimensional\_value (see 4.2.91.3) shall be a tolerance value. When applied to a Numeric\_parameter\_with\_tolerance, the parameter\_value (see 4.2.205.4) shall be a tolerance value. There shall be a qualifier that describes the tolerance context.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.252.

The data associated with a Tolerance\_limit are the following:

- limit\_qualifier.

#### 4.2.325.1 limit\_qualifier

The limit\_qualifier specifies a description of the Tolerance\_limit context.

EXAMPLE '30.5 MAX' or '5 MIN' are Tolerance\_limit examples and the words 'MAX' or 'MIN' are limit\_qualifier examples.

### **4.2.326 Tolerance\_placement**

A `Tolerance_placement` defines how to place a `Dimensional_tolerance` (see 4.2.91) on the geometry of the part. The placement shall define a point of origin and point of termination for the `Dimensional_tolerance`, and the direction the tolerance is to be applied.

The data associated with a `Tolerance_placement` are the following:

- `applied_diredtion`;
- `origin_point`;
- `termination_point`.

#### **4.2.326.1 applied\_diredtion**

The `applied_direction` specifies the direction from the `origin_point` to apply the `Dimensional_tolerance`, the `applied_direction` need not point toward the `origin_point`.

#### **4.2.326.2 origin\_point**

The `origin_point` specifies the point on the Part that defines the starting position for a `Dimensional_`-tolerance.

#### **4.2.326.3 termination\_point**

The `termination_point` specifies the point on the Part that defines the ending position for a `Dimensional_`-tolerance.

### **4.2.327 Tolerance\_range**

A `Tolerance_range` is the upper and lower tolerance range applied directly to a dimension. When applied to a `Dimensional_tolerance`, the `dimensional_value` (see 4.2.91.3) may be a nominal tolerance value. When applied to a `Numeric_parameter_with_tolerance`, the `parameter_value` (see 4.2.205.4) may be a nominal tolerance value.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.253.

The data associated with a `Tolerance_range` are the following:

- `lower_range`;
- `significant_digits`;
- `upper_range`.

#### 4.2.327.1 lower\_range

The lower\_range specifies the lowest allowable value for a dimensional tolerance.

#### 4.2.327.2 significant\_digits

The significant\_digits specifies the number of decimal places indicating the accuracy of the tolerance.

#### 4.2.327.3 upper\_range

The upper\_range specifies the highest allowable value for a dimensional tolerance.

### 4.2.328 Tolerance\_requirement

A Tolerance\_requirement is a type of Customer\_casting\_requirement (see 4.2.62) that is a design requirement for tolerances on a part. The data associated with a Tolerance\_requirement are the following:

- tolerance\_based\_on;
- gaging\_requirement;
- gaging\_fixtures\_available;
- dimension\_method;
- dimension\_specification;
- mismatch.

#### 4.2.328.1 toleranced\_based\_on

The tolerance\_based\_on specifies the tolerance the Tolerance\_requirement is based on. The tolerance shall be either geometric, dimensional or mismatch.

NOTE See 4.2.328.1.1 to 4.2.328.1.3 for the definition of each allowable value for tolerance\_based\_on type.

**4.2.328.1.1 dimensional:** the total amount a specific dimension is permitted to vary, which is the difference between maximum and minimum permitted limits of size.

**4.2.328.1.2 geometric:** the maximum or minimum variation from true geometric form or position that is permitted in manufacture.

**4.2.328.1.3 mismatch:** an imperfection on a cast part is permitted due to inaccurate alignment or a displacement due to shift or dimensional differences between the mould or die components, producing a step on the surface along the parting line.

### 4.2.328.2 gaging\_requirement

The `gaging_requirement` specifies a checking aid to be used for inspection measurements of a cast part. The `gaging_requirement` need not be specified for a particular a `Tolerance_requirement` for the cast part. See 4.3.586 for the application assertion.

EXAMPLE functional gage(s) to be used with open set-up inspection, a positioning fixture(s) for inspection, or tracing(s) when an optical comparator is used for inspection.

### 4.2.328.3 gaging\_fixtures\_available

The `gaging_fixtures_available` specifies a boolean value that indicates if checking aids are available. A value of TRUE specifies that gaging fixtures are available and are required. A value of FALSE specifies that gaging fixtures are not available. See 4.3.586 for the application assertion.

Note If the BOOLEAN has a TRUE value then the attribute `gaging_requirement` is not be OPTIONAL. If the BOOLEAN has a FALSE and the attribute `gaging_requirement` has a value then the checking aids are to be secured.

### 4.2.328.4 dimension\_method

The `dimension_method` specifies the approach for which the tolerance requirement is to be applied.

NOTE See 4.2.328.4.1 to 4.2.328.4.2 for the definition of each allowable value for `dimensional_method` type.

**4.2.328.4.1 specification data:** the dimensional tolerance requirement is determined by a stated specification.

**4.2.328.4.2 drawing data:** the dimensional tolerance requirement is determined by the dimensions and tolerances specified on the design drawing.

### 4.2.328.5 dimension\_specification

The `dimension_specification` specifies the standard for which dimensional tolerances apply. The `dimensional_specification` need not be specified for a particular a `Tolerance_requirement` for the cast part. See 4.3.587 for the application assertion.

### 4.2.328.6 mismatch

The `mismatch` specifies the maximum amount a shift or displacement is permitted at the parting line of a cast part. The `mismatch` need not be specified for a particular a `Tolerance_requirement`. See 4.3.585 for the application assertion.

### 4.2.329 Tolerance\_value

A `Tolerance_value` is the representation of the magnitude of the allowable deviation required for dimensions. These tolerance values may be explicitly defined or may require a specification for definition.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.254.

The data associated with a `Tolerance_value` are the following:

- `defined_value`;

#### 4.2.329.1 defined\_value

The `defined_value` specifies the tolerance deviation value. See 4.3.327, 4.3.328, 4.3.329 and 4.3.330 for the application assertion.

### 4.2.330 Tolerance\_zone

A `Tolerance_zone` is an area where all points of the geometric element that have tolerances shall be contained.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.255.

EXAMPLE A point, line, surface, or plane are examples for geometric elements that have tolerances zones.

The data associated with a `Tolerance_zone` are the following:

- `common_zone`;
- `extended_shape`;
- `form_type`;
- `zone_definition`.

#### 4.2.330.1 common\_zone

The `common_zone` specifies a boolean value that indicates if a `Tolerance_zone` is applied to more than one geometric tolerance. A TRUE value would indicate a common zone.

#### 4.2.330.2 extended\_shape

The `extended_shape` specifies the extension of a feature for the purpose of creating the `Tolerance_zone`. The `extended_shape` need not be specified for a particular `Tolerance_zone`. See 4.3.593 for the application assertion.

ISO 10303-223:2008 (E)

### **4.2.330.3 form\_type**

The `form_type` specifies the shape of the `Tolerance_zone`.

EXAMPLE 'Cylindrical', 'parallelepiped', 'spherical' are examples for `form_type`.

### **4.2.330.4 zone\_definition**

The `zone_definition` specifies the defining boundaries for a `Tolerance_zone`. See 4.3.592 for the application assertion.

### **4.2.331 Tolerance\_zone\_definition**

A `Tolerance_zone_definition` is the boundaries of a `Tolerance_zone`. Each `Tolerance_zone` shall be defined by at least one shape and may be defined with two shapes.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.256.

The data associated with a `Tolerance_zone_definition` are the following:

- `first_element`;
- `second_element`.

#### **4.2.331.1 first\_element**

The `first_element` specifies a geometric shape for defining the boundary for the `Tolerance_zone_definition`. A second element specifies a second geometric shape. See 4.3.594 or the application assertion.

#### **4.2.331.2 second\_element**

The `second_element` specifies the second of two shapes for defining the boundary for the `Tolerance_zone_definition`. The `second_element` need not be specified for a particular `Tolerance_zone_definition`. See 4.3.594 for the application assertion.

### **4.2.332 Tool\_verification**

A `Tool_verification` is the process or procedure used to validate the foundry tool to ensure the customer and metalcaster specifications are met.

The data associated with a `Tool_verification` are the following:

- `based_on_part_design`;
- `core_fit`;



- customer\_requirement;
- verify;
- verify\_clearance.

#### **4.2.332.1 based\_on\_part\_design**

The based\_on\_part\_design specifies the part design version that is used to validate the foundry tooling. See 4.3.597 for the application assertion.

NOTE the foundry tooling may have modified the Design\_part shape to satisfy the metalcaster requirements, but the tooling is to still satisfy the customer form, fit, and function criteria.

#### **4.2.332.2 core\_fit**

The core\_fit specifies the evidence used to ensure that all cores will assemble correctly and fit into the mould properly. The core\_fit need not be specified for a particular a Tool\_verification for a cast part. See 4.3.595 for the application assertion.

NOTE if core\_to\_make is required to produce the cast part the attribute core\_fit is not be OPTIONAL

#### **4.2.332.3 customer\_requirement**

The customer\_requirement specifies one or more customer tolerance requirements to be applied to the foundry tooling that will be used to produce the cast part. See 4.3.599 for the application assertion.

#### **4.2.332.4 verify**

The verify ascertains and certifies that the geometric shape dimensions are as specified by the part design from the customer and the Metal\_casters\_casting\_requirements to produce a cast part. See 4.3.598 or the application assertion.

#### **4.2.332.5 verify\_clearance**

A verify\_clearance validates that one or more of the core prints required to produced the cast part have met the metalcaster requirements. The verify\_clearance need not be specified for a particular a Tool\_verification for a cast part. See4.3.596 for the application assertion.

NOTE if core\_to\_make is required to produce the cast part this attribute is not be OPTIONAL.

### **4.2.333 Tooling**

The Tooling is a type of Part\_version (see 4.2.217) that is a device or implement manufactured to make a non-permanent a mould or is the permanent mould to produce a cast part. Each Tooling shall be either a Die\_cast\_tooling (see 4.2.80), Sand\_casting\_tooling (see 4.2.277), Investment\_casting\_tooling (see 4.2.159), Die\_mould\_tooling (see 4.2.86), Core\_box\_tooling (see 4.2.51).

The data associated with a Tooling are the following:

- date\_last\_used;
- tooling\_designed\_for.

#### **4.2.333.1 date\_last\_used**

The date\_last\_used specifies the day, month and year the Tooling was last used. See 4.3.601 for the application assertion.

#### **4.2.333.2 tooling\_designed\_for**

The tooling\_designed\_for specifies the Casting\_design\_feature (see 4.2.22) required to produce a Cast\_part (see 4.2.19). See 4.3.600 for the application assertion.

### **4.2.334 Tooling\_process**

The Tooling\_process is a type of Casting\_activity (see 4.2.21) that identifies the necessary tool or implement to produce the desired or requested Cast\_part (see 4.2.19).

The data associated with a Tooling\_process are the following:

- cores\_to\_make;
- customer\_requirements;
- number\_of\_impressions;
- part\_geometry;
- process\_method;
- quality\_assurance;
- required\_gating;
- shrinkage\_factor;
- type\_of\_tooling.

#### 4.2.334.1 cores\_to\_make

The `cores_to_make` specifies the `Core_box_tooling` (see 4.2.51) that will be designed to make the required core. There may be more than one `cores_to_make` for a `Tooling_process`. The `cores_to_make` need not be specified for a `Tooling_process`. See 4.3.602 for the application assertion.

#### 4.2.334.2 customer\_requirements

The `customer_requirements` specifies a requirement that shall be satisfied by a tool for any activity or process requiring that tool. There may be more than one `customer_requirements` for a `Tooling_process`. See 4.3.603 for the application assertion.

#### 4.2.334.3 number\_of\_impressions

The `number_of_impressions` specifies the number of cavities or pattern required to a tool.

#### 4.2.334.4 part\_geometry

The `part_geometry` specifies the part design required to manufacture a specified `Part_version` (see 4.2.217) tool. See 4.3.604 for the application assertion.

#### 4.2.334.5 process\_method

The `process_method` specifies the casting process to be used and the tooling needed for that process.

NOTE See 4.2.334.5.1 to 4.2.334.5.3 for the definition of each allowable value for processes.

**4.2.334.5.1 general\_casting:** an ancillary procedure or accessory activity that is used by any metalcaster to produce a casting.

**4.2.334.5.2 non\_permanent\_mould:** a process where the mould is destroyed for each casting produced.

**4.2.334.5.3 permanent\_mould:** a process where the mould is reused for many castings produced.

#### 4.2.334.6 quality\_assurance

The `quality_assurance` specifies the procedure of `Tool_verification` (see 4.2.332) required for each tool used to produce a casting. See 4.3.607 for the application assertion.

#### 4.2.334.7 required\_gating

The `required_gating` specifies the process of designing the tooling for the `Gating_system` (see 4.2.134) required to produce the cast part. See 4.3.605 for the application assertion.

#### **4.2.334.8 shrinkage\_factor**

The `shrinkage_factor` specifies the patternmakers shrink rule or multiplication factor to scale the design part to allow to solidification shrinkage.

#### **4.2.334.9 type\_of\_tooling**

The `type_of_tooling` specifies the required tooling for the casting process to be used to produce the `Cast_part` (see 4.2.19). There may be more than one `type_of_tooling` for a `Tooling_process`. The `type_of_tooling` need not be specified for a `Tooling_process`. See 4.3.606 for the application assertion.

#### **4.2.335 Transition\_feature**

A `Transition_feature` is a type of `Manufacturing_feature` (see 4.2.178) that is a transition area between two surfaces. This feature differs from `Machining_feature` objects in that it requires no orientation for placement.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.258.

#### **4.2.336 Vent**

The `Vent` is a type of `Gating_design_feature` (see 4.2.133) that is a small opening in the mould to facilitate escape of air and gases.

NOTE This application object is harmonized with the vent object from ISO 8062-1:2007, Clause 3.8.

The data associated with a `Vent` are the following:

- `vent_for`;
- `vent_location`;
- `vent_size`.

##### **4.2.336.1 vent\_for**

The `vent_for` specifies the purpose for the `Vent`.

NOTE See 4.2.336.1.1 to 4.2.336.1.3 for the definition of each allowable value for `vent_type`.

**4.2.336.1.1 air release:** to remove entrapped air in a mould, die, or core box.

**4.2.336.1.2 gas release:** to release gases from the mould as molten metal fills the mould cavity.

**4.2.336.1.3 steam:** to allow steam to enter a die or mould.

### 4.2.336.2 vent\_placement

The vent\_placement specifies the location parameters for a Vent. See 4.3.609 for the application assertion.

### 4.2.336.3 vent\_size

The vent\_size specifies size parameters for the vent specified. See 4.3.608 for the application assertion.

### 4.2.337 View\_reference

A View\_reference is the unique identification, within a drawing, of a specific area of interest or view referenced by an activity.

NOTE This application object is harmonized with the same object from ISO 10303-240:2005, Clause 4.2.153.

The data associated with a View\_reference are the following:

- owned\_by;
- sheet;
- view;
- zone.

#### 4.2.337.1 owned\_by

The owned\_by specifies the Design\_reference (see 4.2.77) containing the view. An Design\_reference need not be specified for a particular View\_reference. See 4.3.610 for the application assertion.

#### 4.2.337.2 sheet

The sheet specifies the unique identification, within a drawing, of the logical subdivision of a drawing into multiple presentation areas. These subdivisions correspond to paper sheet sizes for plotting of the drawing. The sheet shall always be specified for a particular View\_reference.

#### 4.2.337.3 view

The view specifies the unique identification, within a drawing, of a two dimensional planar projection of a geometric model from a specified position within its coordinate system. The view need not be specified for a particular View\_reference.

#### 4.2.337.4 zone

The zone specifies the unique identification of an area of interest located within a drawing sheet. The zone need not be specified for a particular View\_reference.

#### 4.2.338 Well

The Well is a type of Gating\_design\_feature (see 4.2.133) that is a basin like reservoir that is located on the drag side of the mould beneath the sprue and riser and at the same center point.

NOTE Figure 20 illustrates a Well and associated Gating\_system.

The data associated with a Well are the following:

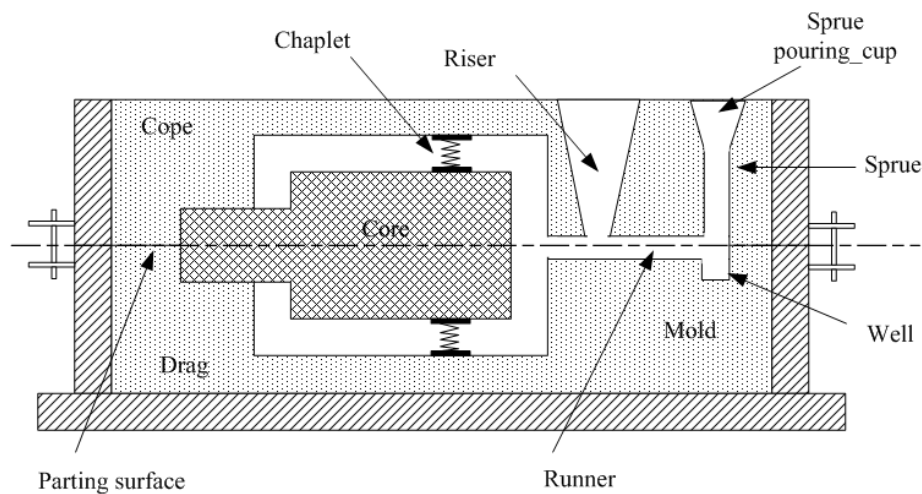
- drag\_surface.
- well\_dimensions.

##### 4.2.338.1 drag\_surface

The drag\_surface specifies the location point on the drag parting surface of the pattern plate for each well. See 4.3.612 for the application assertion.

##### 4.2.338.1.1 well\_dimensions

The well\_dimensions specifies the depth and size of each well. See 4.3.611 for the application assertion.



**Figure 20 — Well and associated Gating\_system**

### 4.2.339 Width\_dimension

A Width\_dimension is a type of Size\_tolerance (see 4.2.304) that specifies the size along a straight line that is referred to as width in the referenced shape.

NOTE This application object is harmonized with the same object from ISO 10303-224:2006, Clause 4.2.261.

The data associated with a Width\_dimension are the following:

— path.

#### 4.2.339.1 path

The path specifies the shape that the tolerance applies to. A Width\_dimension may but need not require a path. See 4.3.613 for the application assertion.

## 4.3 Application assertions

This subclause specifies the application assertions for the Exchange of design and manufacturing product information for cast parts application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

### 4.3.1 Activity to Activity\_execution\_result

Each Activity has the execution\_results defined by zero or one Activity\_execution\_result objects. Each Activity\_execution\_result defines the execution\_results for zero, one, or many Activity objects.

### 4.3.2 Activity to Cast\_part

Each Activity has the output\_item defined by zero or one Cast\_part objects. Each Cast\_part defines the output\_item for zero, one, or many Activity objects.

NOTE This assertion is established through type\_of\_output.

### 4.3.3 Activity to Cast\_part\_with\_rigging

Each Activity has the output\_item defined by zero or one Cast\_part\_with\_rigging objects. Each Cast\_part\_with\_rigging defines the output\_item for zero, one, or many Activity objects.

NOTE This assertion is established through type\_of\_output.

#### **4.3.4 Activity to Design\_reference**

Each Activity has the `graphics_representation` defined by zero, one, or many `Design_reference` objects. Each `Design_reference` defines the `graphics_representation` for zero, one, or many `Activity` objects.

NOTE This assertion is established through `graphics_type_select`.

#### **4.3.5 Activity to Generic\_casting\_resource**

Each Activity has the `consumable_resources` defined by zero, one, or many `Generic_casting_resource` objects. Each `Generic_casting_resource` defines the `consumable_resources` for zero, one, or many `Activity` objects.

#### **4.3.6 Activity to Illustration**

Each Activity has the `graphics_representation` defined by zero, one, or many `Illustration` objects. Each `Illustration` defines the `graphics_representation` for zero, one, or many `Activity` objects.

NOTE This assertion is established through `graphics_type_select`.

#### **4.3.7 Activity to Organization**

Each Activity has the `performed_by_organization_id` defined by zero or one `Organization` objects. Each `Organization` defines the `performed_by_organization_id` for zero, one, or many `Activity` objects.

#### **4.3.8 Activity to Performance\_rate**

Each Activity has the `duration` defined by one or more `Performance_rate` objects. Each `Performance_rate` defines the `used_in` for exactly one `Activity` objects.

#### **4.3.9 Activity to Special\_instruction**

Each Activity has the `activity_information` defined zero, one, or many `Special_instruction`. Each `Special_instruction` defines the `activity_information` for zero, one, or many `Activity` objects.

#### **4.3.10 Activity to Supplemental\_document**

Each Activity has the `reference_to` defined by zero, one, or many `Supplemental_document` objects. Each `Supplemental_document` defines the `reference_to` for zero, one or many `Activity` objects.



### 4.3.11 Activity to View\_reference

Each Activity has the graphics\_representation defined by zero, one, or many View\_reference objects. Each View\_reference defines the graphics\_representation for zero, one, or many Activity objects.

NOTE This assertion is established through graphics\_type\_select.

### 4.3.12 Activity\_execution\_result to Activity\_result\_record

Each Activity\_execution\_result has the activity\_results defined by one or more Activity\_result\_record objects. Each Activity\_result\_record defines the activity\_results for zero, one, or many Activity\_execution\_result objects.

### 4.3.13 Activity\_execution\_result to Sampled\_set

Each Activity\_execution\_result has the part\_measured defined by exactly one Sampled\_set objects. Each Sampled\_set defines the part\_measured for zero, one, or many Activity\_execution\_result objects.

### 4.3.14 Alternate\_activity to Activity

Each Alternate\_activity has the primary\_activity defined by exactly one Activity objects. Each Activity defines the primary\_activity for zero, one or many Alternate\_activity objects.

### 4.3.15 Alternate\_process\_plan to Process\_plan\_version

Each Alternate\_process\_plan has the plan\_substitute defined by exactly one Process\_plan\_version objects. Each Process\_plan\_version defines the plan\_substitute for zero, one or many Alternate\_process\_plan objects.

### 4.3.16 Angular\_dimension\_tolerance to Orientation

Each Angular\_dimension\_tolerance has the plane and direction defined by zero or one Shape\_aspect. Each Orientation defines the plane and direction for zero, one, or many Angular\_dimension\_tolerance objects.

### 4.3.17 Approval to Date

Each Approval has the approval\_date defined by exactly one Date objects. Each Date is the approval\_date for zero, one, or many Approval objects.

### 4.3.18 Approval to Person\_in\_organization

Each Approval has the approval\_authority defined by one or many Person\_in\_organization objects. Each Person\_in\_organization is the approval\_authority for zero, one, or many Approval objects.

#### **4.3.19 Boundary\_condition to Numeric\_parameter**

Each Boundary\_condition has the condition\_value defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the condition\_value for zero, one or many Boundary\_condition objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.20 Boundary\_condition to Property\_relationship**

Each Boundary\_condition has the condition\_value defined by exactly one Property\_relationship objects. Each Property\_relationship defines the condition\_value for zero, one or many Boundary\_condition objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.21 Boundary\_condition to Shape\_aspect**

Each Boundary\_condition has the applies\_to defined by exactly one Shape\_aspect objects. Each Shape\_aspect defines the applies\_to for zero, one or many Boundary\_condition objects.

#### **4.3.22 Cast\_part to Casting\_design\_feature**

Each Cast\_part has the cast\_feature defined by one or more Casting\_design\_feature. Each Casting\_design\_feature defines the cast\_feature for zero, one, or many Cast\_part objects.

#### **4.3.23 Cast\_part\_with\_rigging to Casting\_design\_feature**

Each Cast\_part\_with\_rigging has the cast\_feature defined by one or more Casting\_design\_feature. Each Casting\_design\_feature defines the cast\_feature for zero, one, or many Cast\_part\_with\_rigging objects.

#### **4.3.24 Cast\_part\_with\_rigging to Gating\_system**

Each Cast\_part\_with\_rigging has the rigging defined by exactly one Gating\_system. Each Gating\_system defines the rigging for zero, one, or many Cast\_part\_with\_rigging objects.

#### **4.3.25 Casting\_activity to External\_schema\_definition**

Each Casting\_activity has the micro\_plan\_reference defined by zero or one External\_schema\_reference. Each External\_schema\_reference defines the micro\_plan\_reference for zero, one, or many Casting\_activity objects.

#### **4.3.26 Casting\_design\_feature to Machining\_allowance**

Each Casting\_design\_feature has the machine\_stock defined by zero or one Machining\_allowance. Each Machining\_allowance defines the machine\_stock for zero, one, or many Casting\_design\_feature objects.

#### **4.3.27 Casting\_design\_feature to Numeric\_parameter**

Each Casting\_design\_feature has the draft defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the draft for zero, one, or many Casting\_design\_feature objects.

#### **4.3.28 Casting\_design\_feature to Orientation**

Each Casting\_design\_feature has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Casting\_design\_feature objects.

#### **4.3.29 Casting\_envelope\_requirement to Numeric\_parameter**

Each Casting\_envelope\_requirement has the part\_height defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the part\_height for zero, one, or many Casting\_envelope\_requirement objects.

Each Casting\_envelope\_requirement has the part\_width defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the part\_width for zero, one, or many Casting\_envelope\_requirement objects.

Each Casting\_envelope\_requirement has the part\_length defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the part\_length for zero, one, or many Casting\_envelope\_requirement objects.

#### **4.3.30 Casting\_per\_order to Sampled\_set**

Each Casting\_per\_order has the sampling\_of\_parts defined by exactly one Sampled\_set. Each Sampled\_set defines the sampling\_of\_parts for zero, one, or many Casting\_per\_order objects.

#### **4.3.31 General\_foundry\_process to Equipment\_setup**

Each General\_foundry\_process has the identifies defined by zero or one Equipment\_setup objects. Each Equipment\_setup defines the identifies for zero, one, or many General\_foundry\_process objects.

#### **4.3.32 General\_foundry\_process to Mismatch\_tolerance**

Each General\_foundry\_process has the tolerances defined by zero or one Mismatch\_tolerance objects. Each Mismatch\_tolerance defines the tolerances for zero, one, or many General\_foundry\_process objects.

### **4.3.33 General\_foundry\_process to Numeric\_parameter**

Each General\_foundry\_process has the process\_parameter defined by one or more Numeric\_parameter objects. Each Numeric\_parameter defines the process\_parameter for zero, one, or many General\_foundry\_process objects.

NOTE This assertion is established through property\_select.

### **4.3.34 General\_foundry\_process to Property relationship**

Each General\_foundry\_process has the process\_parameter defined by one or more Property relationship objects. Each Property relationship defines the process\_parameter for zero, one, or many General\_foundry\_process objects.

NOTE This assertion is established through property\_select.

### **4.3.35 General\_foundry\_process to Production\_core\_box**

Each General\_foundry\_process has the core\_to\_be\_made defined by zero or one Production\_core\_box objects. Each Production\_core\_box defines the core\_to\_be\_made for zero, one, or many General\_foundry\_process objects.

### **4.3.36 General\_foundry\_process to Shape\_aspect**

Each General\_foundry\_process has the described\_by defined by zero or one Shape\_aspect objects. Each Shape\_aspect defines the described\_by for zero, one, or many General\_foundry\_process objects.

### **4.3.37 Casting\_service\_data to Corrosion\_service**

Each Casting\_service\_data has the corrosion\_service\_requirement defined by zero or one Corrosion\_service. Each Corrosion\_service defines corrosion\_service\_requirement for zero, one, or many Casting\_service\_data objects.

### **4.3.38 Casting\_service\_data to Descriptive\_parameter**

Each Casting\_service\_data has the comments defined by zero, one, or many Descriptive\_parameter. Each Descriptive\_parameter defines comments for zero, one, or many Casting\_service\_data objects.

Each Casting\_service\_data has the wear\_or\_abrasion\_lubrication defined by zero, one, or many Descriptive\_parameter. Each Descriptive\_parameter defines wear\_or\_abrasion\_lubrication for zero, one, or many Casting\_service\_data objects.

#### 4.3.39 Casting\_service\_data to Elevated\_temperature\_service

Each Casting\_service\_data has the elevated\_temperature\_service defined by zero or one Elevated\_temperature\_service. Each Elevated\_temperature\_service defines elevated\_temperature\_service for zero, one, or many Casting\_service\_data objects.

#### 4.3.40 Casting\_service\_data to Finishing\_or\_machining\_operations

Each Casting\_service\_data has the machining\_and\_finishing\_requirements defined by zero or one Finishing\_or\_machining\_operations. Each Finishing\_or\_machining\_operations defines machining\_and\_finishing\_requirements for zero, one, or many Casting\_service\_data objects.

#### 4.3.41 Casting\_service\_data to Descriptive\_parameter

Each Casting\_service\_data has the subject\_to\_abrasive\_wear\_by defined by zero or one Descriptive\_parameter objects. Each Descriptive\_parameter defines subject\_to\_abrasive\_wear\_by for zero, one, or many Casting\_service\_data objects.

Each Casting\_service\_data has the subject\_to\_wear\_against defined by zero or one Descriptive\_parameter objects. Each Descriptive\_parameter defines subject\_to\_wear\_against for zero, one, or many Casting\_service\_data objects.

#### 4.3.42 Casting\_service\_data to Mechanical\_stress

Each Casting\_service\_data has the mechanical\_stress\_requirements defined by zero or one Mechanical\_stress. Each Mechanical\_stress defines mechanical\_stress\_requirements for zero, one, or many Casting\_service\_data objects.

#### 4.3.43 Casting\_verification to Quality\_acceptance\_report

Each Casting\_verification has the nde defined by zero, one, or many Quality\_acceptance\_report objects. Each Quality\_acceptance\_report defines the nde for zero, one, or many Casting\_verification t objects.

Each Casting\_verification has the destruction defined by zero, one, or many Quality\_acceptance\_report objects. Each Quality\_acceptance\_report defines the destruction for zero, one, or many Casting\_verification t objects.

#### 4.3.44 Casting\_verification to Tolerance\_requirement

Each Casting\_verification has the verify\_with defined by one or more Tolerance\_requirement objects. Each Tolerance\_requirement defines the verify\_with for zero, one, or many Casting\_verification objects.

#### **4.3.45 Casting\_weight\_requirement to Property\_parameter**

Each Casting\_weight\_requirement has the property\_characteristic defined one or more Property\_parameter. Each Property\_parameter defines the property\_characteristic for zero, one, or many Casting\_weight\_requirement objects.

#### **4.3.46 Chaplet to Chaplet\_pad**

Each Chaplet has the pad defined by zero or one Chaplet\_pad. Each Chaplet\_pad defines the pad for zero, one, or many Chaplet objects.

#### **4.3.47 Chaplet to Item\_size**

Each Chaplet has the chaplet\_size defined by exactly one Item\_size. Each Item\_size defines the chaplet\_size for zero, one, or many Chaplet objects.

#### **4.3.48 Chaplet\_pad to Item\_size**

Each Chaplet\_pad has the pad\_size defined by exactly one Item\_size. Each Item\_size defines the pad\_size for zero, one, or many Chaplet\_pad objects.

#### **4.3.49 Checking\_aid\_tool to Date**

Each Checking\_aid\_tool has the revision\_date defined by exactly one Date. Each Date defines the revision\_date for zero, one, or many Checking\_aid\_tool objects.

#### **4.3.50 Chill to Shape\_element**

Each Chill has the chill\_area\_shape defined by zero, one or many Shape\_element objects. Each Shape\_element defines the chill\_area\_shape for zero, one or many Chill objects.

#### **4.3.51 Choke to Item\_size**

Each Choke has the choke\_size defined by exactly one Item\_size. Each Item\_size defines the choke\_size for zero, one, or many Choke objects.

#### **4.3.52 Choke to Location\_element**

Each Choke has the choke\_placement defined by zero, one, or many Location\_element. Each Location\_element defines the choke\_placement for zero, one, or many Choke objects.

#### **4.3.53 Class\_BSU to Supplier\_BSU**

Each Class\_BSU has defined\_by defined by exactly one Supplier\_BSU. Each Supplier\_BSU defines defined\_by for zero, one, or many Class\_BSU objects.

#### **4.3.54 Common\_datum to Datum\_feature**

Each Common\_datum has the element defined by two or more Datum\_feature objects. Each Datum\_feature defines the element for zero, one, or many Common\_datum objects.

#### **4.3.55 Composition\_element\_record to Composition\_element**

Each Composition\_element\_record has the recorded\_usage defined by exactly one Composition\_element objects. Each Composition\_element defines the recorded\_usage for zero, one or many Composition\_element\_record objects.

#### **4.3.56 Composition\_element\_record to Numeric\_parameter**

Each Composition\_element\_record has the actual\_composition\_amount defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the actual\_composition\_amount for zero, one or many Composition\_element\_record objects.

#### **4.3.57 Composition\_inspection to Material**

Each Composition\_inspection has the composition\_result defined by exactly one Material objects. Each Material defines the composition\_result for zero, one or many Composition\_element objects.

#### **4.3.58 Composition\_inspection to Material\_property**

Each Composition\_inspection has the requirement defined by one or more Material\_property objects. Each Material\_property defines the requirement for zero, one or many Composition\_inspection objects.

#### **4.3.59 Compound\_feature to Compound\_feature\_element**

Each Compound\_feature has the element defined by two or more Compound\_feature\_element objects. Each Compound\_feature\_element defines the element for zero, one, or many Compound\_feature objects.

#### **4.3.60 Compound\_feature\_element to Machining\_feature**

Each Compound\_feature\_element has the element defined by exactly one Machining\_feature. Each Machining\_feature defines the element for zero, one, or many Compound\_feature\_element objects.

NOTE This assertion is established through Compound\_feature\_select.

#### **4.3.61 Compound\_feature\_element to Transition\_feature**

Each Compound\_feature\_element has the element defined by exactly one Transition\_feature. Each Transition\_feature defines the element for zero, one, or many Compound\_feature\_element objects.

NOTE This assertion is established through Compound\_feature\_select.

#### **4.3.62 Compound\_feature\_relationship to Compound\_feature\_element**

Each Compound\_feature\_relationship has the successor defined by exactly one Compound\_feature\_element. Each Compound\_feature\_element defines the successor for zero, one, or many Compound\_feature\_relationship objects.

Each Compound\_feature\_relationship has the predecessor defined by exactly one Compound\_feature\_element. Each Compound\_feature\_element defines the predecessor for zero, one, or many Compound\_feature\_relationship objects.

#### **4.3.63 Condition\_or\_assumption to Document\_assignment**

Each Condition\_or\_assumption has the further\_information defined by one or more Document\_assignment objects. Each Document\_assignment defines the further\_information for zero, one or many Condition\_or\_assumption objects.

#### **4.3.64 Condition\_or\_assumption to Numeric\_parameter**

Each Condition\_or\_assumption has the associated\_property defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the associated\_property for zero, one or many Condition\_or\_assumption objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.65 Condition\_or\_assumption to Property\_relationship**

Each Condition\_or\_assumption has the associated\_property defined by exactly one Property\_relationship objects. Each Property\_relationship defines the associated\_property for zero, one or many Condition\_or\_assumption objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.66 Connection\_transition to Numeric\_parameter**

Each Condition\_or\_assumption has the fillet defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the fillet for zero, one or many Condition\_or\_assumption objects.



Each Condition\_or\_assumption has the edge\_round defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the edge\_round for zero, one or many Condition\_or\_assumption objects.

#### **4.3.67 Continuous\_process to Manufacturing\_process**

Each Continuous\_process has the related\_process defined by exactly one Manufacturing\_process objects. Each Manufacturing\_process defines the related\_process for zero, one, or many Continuous\_process objects.

Each Continuous\_process has the relating\_process defined by exactly one Manufacturing\_process objects. Each Manufacturing\_process defines the relating\_process for zero, one, or many Continuous\_process objects.

#### **4.3.68 Controller to Specification**

Each Controller has the controller\_specification defined by exactly one Specification objects. Each Specification defines the controller\_specification for zero, one, or many Controller objects.

#### **4.3.69 Cooling\_port to Item\_size**

Each Cooling\_port has the cooling\_size defined by exactly one Item\_size. Each Item\_size defines the cooling\_size for zero, one, or many Cooling\_port objects.

#### **4.3.70 Cooling\_port to Planar\_element**

Each Cooling\_port has the port\_location defined by exactly one Planar\_element. Each Planar\_element defines the port\_location for zero, one, or many Cooling\_port objects.

#### **4.3.71 Core to Vent**

Each Core has the gas\_vent defined by zero, one, or many Vent. Each Vent defines the gas\_vent for zero, one, or many Core objects.

#### **4.3.72 Core to Sand\_mould**

Each Core has the mould\_shape\_definition defined by zero, one, or many Sand\_mould. Each Sand\_mould defines the mould\_shape\_definition for zero, one, or many Core objects.

#### **4.3.73 Core\_assembly to Descriptive\_parameter**

Each Core\_assembly has the hard\_form defined by zero, one, or many Descriptive\_parameter. Each Descriptive\_parameter defines hard\_form for zero, one, or many Core\_assembly objects.

#### **4.3.74 Core\_assembly to Proof\_report**

Each Core\_assembly has the soft\_copy defined by zero, one, or many Proof\_report. Each Proof\_report defines soft\_copy for zero, one, or many Core\_assembly objects.

#### **4.3.75 Core\_box\_tooling to Production\_core\_box**

Each Core\_box\_tooling has the core\_box\_equipment defined by zero or one Production\_core\_box. Each Production\_core\_box defines the core\_box\_equipment for zero, one, or many Core\_box\_tooling objects.

#### **4.3.76 Core\_box\_vent to Equipment\_size**

Each Core\_box\_vent has the vent\_dimensions defined by zero or one Equipment\_size. Each Equipment\_size defines the vent\_dimensions for zero, one, or many Core\_box\_vent objects.

#### **4.3.77 Core\_box\_vent to Shape\_element**

Each Core\_box\_vent has the vent\_dimensions defined by zero or one Shape\_element. Each Shape\_element defines the vent\_dimensions for zero, one, or many Core\_box\_vent objects.

#### **4.3.78 Core\_conveyer\_system to Equipment\_size**

Each Core\_conveyer\_system has the core\_box\_skid defined by exactly one Equipment\_size. Each Equipment\_size defines the core\_box\_skid for zero, one, or many Core\_conveyer\_system objects.

Each Core\_conveyer\_system has the core\_plug defined by zero or one Equipment\_size. Each Equipment\_size defines the core\_plug for zero, one, or many Core\_conveyer\_system objects.

#### **4.3.79 Core\_machine to Ejector\_equipment**

Each Core\_machine has the ejection\_definition defined by zero, one or many Ejector\_equipment. Each Ejector\_equipment defines the ejection\_definition for zero, one, or many Core\_machine objects.

#### **4.3.80 Core\_machine to Machine\_mounting**

Each Core\_machine has the tool\_mounting\_definition defined by zero or one Machine\_mounting. Each Machine\_mounting defines the tool\_mounting\_definition for zero, one, or many Core\_machine objects.

#### **4.3.81 Core\_machine to Core\_box\_vent**

Each Core\_machine has the vent\_definition defined by zero or one Core\_box\_vent. Each Core\_box\_vent defines the vent\_definition for zero, one, or many Core\_machine objects.

#### **4.3.82 Core\_print to Drafted\_surface**

Each Core\_print has the draft defined by exactly one Drafted\_surface. Each Drafted\_surface defines the draft for zero, one, or many Core\_print objects.

#### **4.3.83 Core\_print to Shape\_element**

Each Core\_print has the feature defined by one or many Shape\_element. Each Shape\_element defines the feature for zero, one, or many Core\_print objects.

#### **4.3.84 Core\_print to Item\_size**

Each Core\_print has the core\_print\_dimensions defined by exactly one Item\_size. Each Item\_size defines the core\_print\_dimensions for zero, one, or many Core\_print objects.

#### **4.3.85 Corrosion\_service to Descriptive\_parameter**

Each Corrosion\_service has the corrosive\_environment defined by zero or one Descriptive\_parameter objects. Each Descriptive\_parameter defines the corrosive\_environment for zero, one, or many Corrosion\_service objects.

Each Corrosion\_service has the corrosive\_material defined by zero or one Descriptive\_parameter objects. Each Descriptive\_parameter defines the corrosive\_material for zero, one, or many Corrosion\_service objects.

#### **4.3.86 Corrosion\_service to Numeric\_parameter**

Each Corrosion\_service has the ph\_value defined by zero or one Numeric\_parameter objects. Each Numeric\_parameter defines the ph\_value for zero, one, or many Corrosion\_service objects.

Each Corrosion\_service has the temperature defined by zero or one Numeric\_parameter objects. Each Numeric\_parameter defines the temperature for zero, one, or many Corrosion\_service objects.

#### **4.3.87 Customer\_casting\_requirement to Illustration**

Each Customer\_casting\_requirement has the illustration defined by zero, one, or many Illustration objects. Each Illustration defines the illustration for zero, one, or many Customer\_casting\_requirement objects.

#### **4.3.88 Customer\_casting\_requirement to Shape\_aspect**

Each Customer\_casting\_requirement has the applied\_to defined by zero, one, or many Shape\_aspect objects. Each Shape\_aspect defines the applied\_to for zero, one, or many Customer\_casting\_requirement objects.

### **4.3.89 Customer\_casting\_requirement to Specification**

Each Customer\_casting\_requirement has the referenced\_from defined by zero, one, or many Specification objects. Each Specification defines the referenced\_from for zero, one, or many Customer\_casting\_requirement objects.

### **4.3.90 Customer\_order to Date**

Each Customer\_order has the delivery\_date defined by exactly one Date. Each Date defines the delivery\_date for zero, one, or many Customer\_order objects.

### **4.3.91 Customer\_order to First\_article**

Each Customer\_order has the first\_article\_requirements defined by one or more First\_article. Each First\_article defines the first\_article\_requirements for zero, one, or many Customer\_order objects.

### **4.3.92 Customer\_order to Inspection\_plan**

Each Customer\_order has the inspection\_procedure defined by exactly one Inspection\_plan. Each Inspection\_plan defines the inspection\_procedure for zero, one, or many Customer\_order objects.

### **4.3.93 Customer\_order to Ordered\_part**

Each Customer\_order has the quantity\_ordered defined by one or many Ordered\_part objects. Each Ordered\_part defines the quantity\_ordered for zero or one Customer\_order.

### **4.3.94 Customer\_order to Person\_in\_organization**

Each Customer\_order has the customer defined by exactly one Person\_in\_organization. Each Person\_in\_organization defines the customer for zero, one, or many Customer\_order objects.

### **4.3.95 Customer\_order to Project\_order**

Each Customer\_order has the initiated\_order defined by zero, one, or many Project\_order objects. Each Project\_order defines the initiated\_order for zero or one Customer\_order.

### **4.3.96 Customer\_order to Request\_for\_quotation**

Each Customer\_order has the RFQ defined by zero or one Request\_for\_quotation. Each Request\_for\_quotation defines the RFQ for zero, one, or many Customer\_order objects.

#### **4.3.97 Customer\_order to Simulation\_report**

Each Customer\_order has the simulation\_run\_result defined by zero or one Simulation\_report. Each Simulation\_report defines the simulation\_run\_result for zero, one, or many Customer\_order objects.

#### **4.3.98 Customer\_simulation to Customer\_design\_deficiency\_report**

Each Customer\_simulation has the defect\_results defined by zero, one, or many Customer\_design\_deficiency\_report. Each Customer\_design\_deficiency\_report defines the defect\_results for zero, one, or many Customer\_simulation objects.

#### **4.3.99 Customer\_customer\_tool\_design\_report to Tooling**

Each Customer\_customer\_tool\_design\_report has the designed\_tool defined by exactly one Tooling. Each Tooling defines the designed\_tool for zero, one, or many Customer\_customer\_tool\_design\_report objects.

#### **4.3.100 Data\_cure to Curve\_point**

Each Data\_cure has the data\_points defined by one or more Curve\_point. Each Curve\_point defines the data\_points for zero, one, or many Data\_cure objects.

#### **4.3.101 Datum\_feature to Material\_condition\_modifier**

Each Datum has the modifier defined by zero or one Material\_condition\_modifier. Each Material\_condition\_modifier defines the modifier for zero, one, or many Datum objects.

#### **4.3.102 Datum\_feature to Datum\_target\_set**

Each Datum\_feature has the datum\_representation defined by exactly one Datum\_target\_set. Each Datum\_target\_set defines the datum\_representation for zero, one, or many Datum\_feature objects.

NOTE This assertion is established through Datum\_representation\_select.

#### **4.3.103 Datum\_feature to Shape\_element**

Each Datum\_feature has the datum\_representation defined by exactly one Shape\_element. Each Shape\_element defines the datum\_representation for zero, one, or many Datum\_feature objects.

NOTE This assertion is established through Datum\_representation\_select.

#### **4.3.104 Datum\_target\_set to Datum\_target**

Each Datum\_target\_set has the target\_shape defined by one or many Datum\_target objects. Each Datum\_target defines the target\_shape for zero, one, or many Datum\_target\_set objects.

#### **4.3.105 Defect\_prediction to Shape\_aspect**

Each Defect\_prediction has the defect\_shape defined by exactly one Shape\_aspect. Each Shape\_aspect defines the defect\_shape for zero, one, or many Defect\_prediction objects.

#### **4.3.106 Defect\_prediction to Location\_element**

Each Defect\_prediction has the location defined by exactly one Location\_element. Each Location\_element defines the location for zero, one, or many Defect\_prediction objects.

#### **4.3.107 Design\_exception\_notice to Date**

Each Design\_exception\_notice has the issuing\_date defined by exactly one Date objects. Each Date defines the issuing\_date for exactly one Design\_exception\_notice.

#### **4.3.108 Design\_exception\_notice to Engineering\_change\_proposal**

Each Design\_exception\_notice has the issues defined by zero, one, or many Engineering\_change\_proposal objects. Each Engineering\_change\_proposal defines the formalizes for exactly one Design\_exception\_notice.

#### **4.3.109 Design\_exception\_notice to Part\_version**

Each Design\_exception\_notice has the discrepant\_part defined by one or many Part\_version objects. Each Part\_version defines the discrepant\_part for zero, one, or many Design\_exception\_notice objects.

#### **4.3.110 Design\_part to Machine\_design\_feature**

Each Design\_part has the design\_feature defined by zero, one, or many Machine\_design\_feature objects. Each Machine\_design\_feature defines the design\_feature for zero, one or many Design\_part objects.

#### **4.3.111 Design\_reference to Document\_file**

Each Design\_reference has the drawing\_data\_file defined by exactly one Document\_file objects. Each Document\_file defines the drawing\_data\_file for zero, one, or many Design\_reference objects.

#### **4.3.112 Design\_reference to Activity**

Each Design\_reference has the identifies defined by zero, one, or many Activity objects. Each Activity object defines the identifies for zero, one, or many Design\_reference objects.

#### 4.3.113 Design\_reference to Part\_version

Each Design\_reference has the documents\_version defined by exactly one Part\_version objects. Each Part\_version object defines the documents\_version for zero, one, or many Design\_reference objects.

#### 4.3.114 Die\_cast\_master to Production\_die\_cast\_mould

Each Die\_cast\_master has the male\_component defined by one or more Production\_die\_cast\_mould. Each Production\_die\_cast\_mould defines the male\_component for zero, one, or many Die\_cast\_master objects.

Each Die\_cast\_master has the female\_component defined by one or more Production\_die\_cast\_mould. Each Production\_die\_cast\_mould defines the female\_component for zero, one, or many Die\_cast\_master objects.

#### 4.3.115 Die\_cast\_master to Machining\_allowance

Each Die\_cast\_master has the machining\_allowance\_on\_master\_tooling defined by one or more Machining\_allowance. Each Machining\_allowance defines the machining\_allowance\_on\_master\_tooling for zero, one, or many Die\_cast\_master objects.

#### 4.3.116 Die\_cast\_tooling to Material

Each Die\_cast\_tooling has the material\_definition defined by exactly one Material objects. Each Material defines the material\_definition for one or more Die\_cast\_tooling objects.

#### 4.3.117 Die\_cast\_tooling to Production\_die\_cast\_mould

Each Die\_cast\_tooling has the die\_equipment defined by two or more Production\_die\_cast\_mould. Each Production\_die\_cast\_mould defines the die\_equipment for zero, one, or many Die\_cast\_tooling objects.

#### 4.3.118 Die\_cast\_tooling to In\_facility\_location

Each Die\_cast\_tooling has the tool\_location defined by two or more In\_facility\_location. Each In\_facility\_location the tool\_location for zero, one, or many Die\_cast\_tooling objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### 4.3.119 Die\_cast\_tooling to Person\_in\_organization

Each Die\_cast\_tooling has the tool\_location defined by two or more Person\_in\_organization. Each Person\_in\_organization the tool\_location for zero, one, or many Die\_cast\_tooling objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### **4.3.120 Die\_clamping to Orientation**

Each Die\_clamping has the die\_clamping\_positions defined by one or more Orientation. Each Orientation defines the die\_clamping\_positions for zero, one, or many Die\_clamping objects.

#### **4.3.121 Die\_core to Drafted\_surface**

Each Die\_core has the draft defined by exactly one Drafted\_surface. Each Drafted\_surface defines the draft for zero, one, or many Die\_core objects.

#### **4.3.122 Die\_core to Item\_size**

Each Die\_core has the guide\_pin defined by zero, one, or many Item\_size. Each Item\_size defines the guide\_pin for zero, one, or many Die\_core objects.

#### **4.3.123 Die\_core to Parting\_surface**

Each Die\_core has the parting defined by exactly one Parting\_surface objects. Each Parting\_surface defines the parting for zero, one, or many Die\_core objects.

#### **4.3.124 Die\_core to Shape\_aspect**

Each Die\_core has the shape defined by exactly one Shape\_aspect. Each Shape\_aspect defines the shape for zero, one, or many Die\_core objects.

#### **4.3.125 Die\_design\_feature to Orientation**

Each Die\_design\_feature has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Die\_design\_feature objects.

#### **4.3.126 Die\_master to Production\_die\_mould**

Each Die\_master has the female\_component defined by exactly one Production\_die\_mould. Each Production\_die\_mould defines the female\_component for zero, one, or many Die\_master objects.

Each Die\_master has the male\_component defined by exactly one Production\_die\_mould. Each Production\_die\_mould defines the male\_component for zero, one, or many Die\_master objects.

#### **4.3.127 Die\_master to Machining\_allowance**

Each Die\_master has the machining\_allowance\_on\_tooling defined by zero or one Machining\_allowance. Each Machining\_allowance defines the machining\_allowance\_on\_tooling for zero, one, or many Die\_master objects.



#### **4.3.128 Die\_mould\_tooling to Material**

Each Die\_mould\_tooling has the material\_definition defined by exactly one Material objects. Each Material defines the material\_definition for one or more Die\_mould\_tooling objects.

#### **4.3.129 Die\_mould\_tooling to In\_facility\_location**

Each Die\_mould\_tooling has the tooling\_location defined by one or more In\_facility\_location. Each In\_facility\_location the tooling\_location for zero, one, or many Die\_mould\_tooling objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### **4.3.130 Die\_mould\_tooling to Person\_in\_organization**

Each Die\_mould\_tooling has the tooling\_location defined by one or more Person\_in\_organization. Each Person\_in\_organization the tooling\_location for zero, one, or many Die\_mould\_tooling objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### **4.3.131 Die\_mould\_tooling to Production\_die\_mould**

Each Die\_mould\_tooling has the die\_equipment defined by two or more Production\_die\_mould. Each Production\_die\_mould defines the die\_equipment for zero, one, or many Die\_mould\_tooling objects.

#### **4.3.132 Die\_mould\_vent to Ejector\_pins**

Each Die\_mould\_vent has the ejector\_pin\_definition defined by one or more Ejector\_pins. Each Ejector\_pins defines the ejector\_pin\_definition for zero, one, or many Die\_mould\_vent objects.

#### **4.3.133 Dimensional\_measurement\_inspection to External\_schema\_definition**

Each Dimensional\_measurement\_inspection has the iso10303\_219\_data defined by exactly one External\_schema\_definition objects. Each External\_schema\_definition defines the iso10303\_219\_data for zero, one or many Dimensional\_measurement\_inspection objects.

#### **4.3.134 Dimensional\_tolerance to Tolerance\_value**

Each Dimensional\_tolerance has the limit defined by zero or one Tolerance\_value. Each Tolerance\_value defines the limit for zero, one, or many Dimensional\_tolerance objects.

#### **4.3.135 Dimensional\_tolerance to Tolerance\_placement**

Each Dimensional\_tolerance has the placement defined by zero or one Tolerance\_placement. Each Tolerance\_placement defines the limit for zero, one, or many Dimensional\_tolerance objects.

#### **4.3.136 Distance\_along\_curve\_tolerance to Shape\_aspect**

Each Distance\_along\_curve\_tolerance has a path defined by exactly one Shape\_aspect. Each Shape\_aspect defines the path for zero, one, or many Distance\_along\_curve\_tolerance objects.

#### **4.3.137 Document\_assignment to Document\_file**

Each Document\_assignment has the assigned\_document defined by exactly one Document\_file objects. Each Document\_file defines the assigned\_document for zero, one, or many Document\_assignment objects.

#### **4.3.138 Document\_file to Document\_file\_properties**

Each Document\_file has the document\_properties defined by zero or one Document\_file\_properties objects. Each Document\_file\_properties defines the document\_properties for zero, one, or many Document\_file objects.

#### **4.3.139 Document\_file to External\_file\_identification**

Each Document\_file has the file\_location defined by zero, one or many External\_file\_identification objects. Each External\_file\_identification defines the file\_location for zero, one, or many Document\_file objects.

#### **4.3.140 Document\_file\_relationship to Document\_file**

Each Document\_file\_relationship has the previous\_file defined by exactly one Document\_file objects. Each Document\_file defines the previous\_file for zero, one, or many Document\_file\_relationship objects.

Each Document\_file\_relationship has the succeeding\_file defined by exactly one Document\_file objects. Each Document\_file defines the succeeding\_file for zero, one, or many Document\_file\_relationship objects.

#### **4.3.141 Draft\_angle\_surface to Numeric\_parameter**

Each Draft\_angle\_surface has the draft\_amount defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the draft\_amount for zero, one, or many Draft\_angle\_surface objects.

#### **4.3.142 Draft\_angle\_surface to Face\_shape\_element**

Each Draft\_angle\_surface has the shape\_element defined by exactly one Face\_shape\_element. Each Face\_shape\_element defines the shape\_element for zero, one, or many Draft\_angle\_surface objects.

NOTE This assertion is established through the Shape\_element\_select.

#### 4.3.143 Draft\_angle\_surface to Planar\_element

Each Draft\_angle\_surface has the shape\_element defined by exactly one Planar\_element. Each Planar\_element defines the shape\_element for zero, one, or many Draft\_angle\_surface objects.

NOTE This assertion is established through the Shape\_element\_select.

#### 4.3.144 Ejector\_box to Ejector\_plate

Each Ejector\_box has a plate\_definition defined by one or more Ejector\_plate. Each Ejector\_plate defines the plate\_definition for zero, one, or many Ejector\_box objects.

#### 4.3.145 Ejector\_box to Item\_size

Each Ejector\_box has a box\_size defined by exactly one Item\_size. Each Item\_size defines the box\_size for zero, one, or many Ejector\_box objects.

#### 4.3.146 Ejector\_box to Production\_core\_box

Each Ejector\_box has an oriented\_to\_movable\_die defined by exactly one Production\_core\_box. Each Production\_core\_box defines the oriented\_to\_movable\_die for zero, one, or many Ejector\_box objects.

NOTE This assertion is established through the die\_select.

#### 4.3.147 Ejector\_box to Production\_die\_mould

Each Ejector\_box has an oriented\_to\_movable\_die defined by exactly one Production\_die\_mould. Each Production\_die\_mould defines the oriented\_to\_movable\_die for zero, one, or many Ejector\_box objects.

NOTE This assertion is established through the die\_select.

#### 4.3.148 Ejector\_box to Production\_die\_cast\_mould

Each Ejector\_box has an oriented\_to\_movable\_die defined by exactly one Production\_die\_cast\_mould. Each Production\_die\_cast\_mould defines the oriented\_to\_movable\_die for zero, one, or many Ejector\_box objects.

NOTE This assertion is established through the die\_select.

#### 4.3.149 Ejector\_box to Production\_investment\_cast\_mould

Each Ejector\_box has an oriented\_to\_movable\_die defined by exactly one Production\_investment\_cast\_mould. Each Production\_investment\_cast\_mould defines the oriented\_to\_movable\_die for zero, one, or many Ejector\_box objects.

NOTE This assertion is established through the die\_select.

#### **4.3.150 Ejector\_design\_feature to Orientation**

Each Ejector\_design\_feature has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Ejector\_design\_feature objects.

#### **4.3.151 Ejector\_equipment to Ejector\_system**

Each Ejector\_equipment has the ejection\_characteristic defined by exactly one Ejector\_system. Each Ejector\_system defines the ejection\_characteristic for zero, one, or many Ejector\_equipment objects.

#### **4.3.152 Ejector\_pins to Machining\_feature**

Each Ejector\_pins has a thread defined by exactly one Machining\_feature. Each Machining\_feature defines the thread for zero, one, or many Ejector\_pins objects.

#### **4.3.153 Ejector\_pins to Item\_size**

Each Ejector\_pins has a pin\_dimensions defined by exactly one Item\_size. Each Item\_size defines the pin\_dimensions for zero, one, or many Ejector\_pins objects.

#### **4.3.154 Ejector\_pins to Pin\_tips**

Each Ejector\_pins has a tip\_definition defined by exactly one Pin\_tips. Each Pin\_tips defines the tip\_definition for zero, one, or many Ejector\_pins objects.

#### **4.3.155 Ejector\_plate to Ejector\_pins**

Each Ejector\_plate has a ejector\_pin\_definitions defined by one or more Ejector\_pins. Each Ejector\_pins defines the ejector\_pin\_definitions for zero, one, or many Ejector\_plate objects.

#### **4.3.156 Ejector\_plate to Item\_size**

Each Ejector\_plate has a plate\_shape defined by zero or one Item\_size. Each Item\_size defines the plate\_shape for zero, one, or many Ejector\_plate objects.

#### **4.3.157 Ejector\_plate to Numeric\_parameter**

Each Ejector\_plate has a plate\_height defined by exactly three Numeric\_parameter. Each Numeric\_parameter defines the plate\_height for zero, one, or many Ejector\_plate objects.

#### **4.3.158 Ejector\_plate to Reset\_or\_return\_pins**

Each Ejector\_plate has a reset\_push\_pin\_definitions defined by one or more Reset\_or\_return\_pins. Each Reset\_or\_return\_pins defines the reset\_push\_pin\_definitions for zero, one, or many Ejector\_plate objects.

#### **4.3.159 Ejector\_plate to Stop\_pin**

Each Ejector\_plate has a stop\_pin\_definition defined by zero, one, or many Stop\_pin. Each Stop\_pin defines the stop\_pin\_definition for zero, one, or many Ejector\_plate objects.

#### **4.3.160 Ejector\_system to Ejector\_plate**

Each Ejector\_system has a plate\_definition defined by one or more Ejector\_plate. Each Ejector\_plate defines the plate\_definition for zero, one, or many Ejector\_system objects.

#### **4.3.161 Elevated\_temperature\_service to Descriptive\_parameter**

Each Elevated\_temperature\_service has a service\_descriptions defined by zero, one, or many Descriptive\_parameter. Each Descriptive\_parameter defines the service\_descriptions for zero, one, or many Elevated\_temperature\_service objects.

#### **4.3.162 Elevated\_temperature\_service to Numeric\_parameter**

Each Elevated\_temperature\_service has a maximum\_temperature defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the maximum\_temperature for zero, one, or many Elevated\_temperature\_service objects.

Each Elevated\_temperature\_service has a maximum\_cyclic\_temperature defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the maximum\_cyclic\_temperature for zero, one, or many Elevated\_temperature\_service objects.

Each Elevated\_temperature\_service has a minimum\_cyclic\_temperature defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the minimum\_cyclic\_temperature for zero, one, or many Elevated\_temperature\_service objects.

#### **4.3.163 Engineering\_change\_order to Part\_version**

Each Engineering\_change\_order has the new\_version defined by one or more Part\_version objects. Each Part\_version defines the new\_version by zero, one or many Engineering\_change\_order.

#### **4.3.164 Engineering\_change\_proposal to Engineering\_change\_order**

Each Engineering\_change\_proposal has the incorporated\_proposal defined by one or more Engineering\_change\_order objects. Each Engineering\_change\_order defines the approved\_proposal for exactly one Engineering\_change\_proposal.

#### **4.3.165 Equipment to Controller**

Each Equipment has the controlled\_by defined zero or one Controller. Each Controller defines the controlled\_by for zero, one, or many Equipment objects.

#### **4.3.166 Equipment to In\_facility\_location**

Each Equipment has the location defined by exactly one In\_facility\_location. Each In\_facility\_location defines the location for zero, one, or many Equipment objects.

#### **4.3.167 Equipment\_process to Equipment**

Each Equipment\_process has the required\_edquipment defined exactly one Equipment. Each Equipment defines the required\_edquipment for zero, one, or many Equipment\_process objects.

#### **4.3.168 Equipment\_setting\_record to Equipment\_usage**

Each Equipment\_setting\_record has the required\_edquipment defined exactly one Equipment. Each Equipment\_usage defines the required\_edquipment for zero, one, or many Equipment\_setting\_record objects.

#### **4.3.169 Equipment\_setting\_record to Property\_parameter**

Each Equipment\_setting\_record has the actual\_machine\_setting defined one or more Property\_parameter. Each Property\_parameter defines the actual\_machine\_setting for zero, one, or many Equipment\_setting\_record objects.

#### **4.3.170 Equipment\_setup to Equipment**

Each Equipment\_setup has the identifies defined exactly one Equipment. Each Equipment defines the identifies for zero, one, or many Equipment\_setup objects.

#### **4.3.171 Equipment\_size to Numeric\_parameter**

Each Equipment\_size has the item\_height defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the item\_height for zero, one, or many Equipment\_size objects.

Each Equipment\_size has the item\_length defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the item\_length for zero, one, or many Equipment\_size objects.

Each Equipment\_size has the item\_width defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the item\_width for zero, one, or many Equipment\_size objects.

Each Equipment\_size has the item\_diameter defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the item\_diameter for zero, one, or many Equipment\_size objects.

Each Equipment\_size has the item\_draft defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the item\_draft for zero, one, or many Equipment\_size objects.

#### **4.3.172 Equipment\_usage to Property\_parameter**

Each Equipment\_usage has the equipment\_usage\_context defined exactly one Property\_parameter. Each Property\_parameter defines the equipment\_usage\_context for zero, one, or many Equipment\_usage objects.

#### **4.3.173 Equipment\_usage to Equipment**

Each Equipment\_usage has the equipment\_parameter defined exactly one Equipment. Each Equipment defines the equipment\_parameter for zero, one, or many Equipment\_usage objects.

#### **4.3.174 External\_schema\_definition to Document\_assignment**

Each External\_schema\_definition has the reference\_schema defined by exactly one Document\_assignment objects. Each Document\_assignment defines the reference\_schema for zero, one, or many External\_schema\_definition objects.

#### **4.3.175 Externally\_defined\_representation to Library\_part\_assignment**

Each Externally\_defined\_representation has the identified\_by defined by exactly one Library\_part\_assignment objects. Each Library\_part\_assignment defines the identified\_by for zero, one, or many Externally\_defined\_representation objects.

#### **4.3.176 Externally\_defined\_size\_dimension to Shape\_aspect**

Each Externally\_defined\_size\_dimension has the applied\_shape defined zero or one Shape\_aspect. Each Shape\_aspect defines the applied\_shape for zero, one, or many Externally\_defined\_size\_dimension objects.

#### **4.3.177 Externally\_defined\_size\_dimension to Document\_assignment**

Each Externally\_defined\_size\_dimension has the tolerance\_definition defined by exactly one Document\_assignment objects. Each Document\_assignment defines the tolerance\_definition for zero, one, or many Externally\_defined\_size\_dimension objects.

#### **4.3.178 Face\_shape\_element\_relationship to Face\_shape\_element**

Each Face\_shape\_element\_relationship has the predecessor defined by exactly one Face\_shape\_element. Each Face\_shape\_element defines the predecessor for zero, one, or many Face\_shape\_element\_relationship objects.

Each Face\_shape\_element\_relationship has the successor defined by exactly one Face\_shape\_element. Each Face\_shape\_element defines the successor for zero, one, or many Face\_shape\_element\_relationship objects.

#### **4.3.179 Feeders to Connection\_transition**

Each Feeders has the transition defined by exactly one Connection\_transition objects. Each Connection\_transition defines the transition for zero, one, or many Feeders objects.

#### **4.3.180 Feeders to Risers**

Each Feeders has the runner\_connection defined by zero, one, or many Risers objects. Each Risers defines the runner\_connection for zero, one, or many Feeders objects.

#### **4.3.181 Feeders to Item\_size**

Each Feeders has a parametric\_shape defined by zero or one Item\_size. Each Item\_size defines the parametric\_shape for zero, one, or many Feeders objects.

#### **4.3.182 Filter to Item\_size**

Each Filter has the filter\_size defined by exactly one Item\_size. Each Item\_size defines the filter\_size for zero, one, or many Filter objects.

#### **4.3.183 Finishing\_or\_machining\_operations to Descriptive\_parameter**

Each Finishing\_or\_machining\_operations has the comments defined by zero or one Descriptive\_parameter. Each Descriptive\_parameter defines the comments for zero, one, or many Finishing\_or\_machining\_operations objects.

#### **4.3.184 Finishing\_or\_machining\_operations to Machining\_activity**

Each Finishing\_or\_machining\_operations has the finishing\_operations defined by zero, one, or many Machining\_activity. Each Machining\_activity defines the finishing\_operations for zero, one, or many Finishing\_or\_machining\_operations objects.

Each Finishing\_or\_machining\_operations has the machining\_operation defined by zero, one, or many Machining\_activity. Each Machining\_activity defines the machining\_operation for zero, one, or many Finishing\_or\_machining\_operations objects.

#### **4.3.185 First\_article to Approval**

Each First\_article has the customer\_approval defined by exactly one Approval object. Each Approval defines the customer\_approval for zero, one, or many First\_article objects.

#### **4.3.186 First\_article to Date**

Each First\_article has the delivery\_date defined by exactly one Date object. Each Date defines the delivery\_date for zero, one, or many First\_article objects.



**4.3.187 First\_article to Master\_sample**

Each First\_article has the customer\_approved defined by exactly one Master\_sample object. Each Master\_sample defines the customer\_approved for zero, one, or many First\_article objects.

**4.3.188 Flask to Numeric\_parameter**

Each Flask has a cope\_height defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the cope\_height for zero, one, or many Flask objects.

Each Flask has a flask\_height defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the flask\_height for zero, one, or many Flask objects.

Each Flask has a flask\_length defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the flask\_length for zero, one, or many Flask objects.

Each Flask has a drag\_height defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the drag\_height for zero, one, or many Flask objects.

**4.3.189 Flask to Descriptive\_parameter**

Each Flask has a type\_of\_flask defined by exactly one Descriptive\_parameter. Each Descriptive\_parameter defines the type\_of\_flask for zero, one, or many Flask objects.

**4.3.190 Flask to Pin\_center**

Each Flask has a pin\_center\_definition defined by zero or one Pin\_center. Each Pin\_center defines the pin\_center\_definition for zero, one, or many Flask objects.

**4.3.191 Flask to Sand\_mould**

Each Flask has a generates defined by zero or one Sand\_mould. Each Sand\_mould defines the generates for zero, one, or many Flask objects.

**4.3.192 Flaskless to Mould\_box**

Each Flaskless has a mould\_made\_by defined by zero, one, or many Mould\_box. Each Mould\_box defines the mould\_made\_by for zero, one, or many Flaskless objects.

**4.3.193 Gating\_design\_feature to Orientation**

Each Gating\_design\_feature has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Gating\_design\_feature objects.

#### **4.3.194 Gating\_system to Ingate**

Each Gating\_system has a gating\_components defined by one or more Ingate. Each Ingate defines the pin\_center\_definition for zero, one, or many Gating\_system objects.

NOTE This assertion is established through the Gating\_component\_select.

#### **4.3.195 Gating\_system to Risers**

Each Gating\_system has a gating\_components defined by one or more Risers. Each Risers defines the pin\_center\_definition for zero, one, or many Gating\_system objects.

NOTE This assertion is established through the Gating\_component\_select.

#### **4.3.196 Gating\_system to Runners**

Each Gating\_system has a gating\_components defined by one or more Runners. Each Runners defines the pin\_center\_definition for zero, one, or many Gating\_system objects.

NOTE This assertion is established through the Gating\_component\_select.

#### **4.3.197 Gating\_system to Sprue**

Each Gating\_system has a gating\_components defined by one or more Sprue. Each Sprue defines the pin\_center\_definition for zero, one, or many Gating\_system objects.

NOTE This assertion is established through the Gating\_component\_select.

#### **4.3.198 General\_rib to Numeric\_parameter**

Each General\_rib has a part\_rib\_fillet defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the part\_rib\_fillet for zero, one, or many General\_rib objects.

Each General\_rib has a wall\_to\_top\_radius defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the wall\_to\_top\_radius for zero, one, or many General\_rib objects.

Each General\_rib has a wall\_to\_floor\_fillet defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the wall\_to\_floor\_fillet for zero, one, or many General\_rib objects.

#### **4.3.199 General\_rib to Planar\_element**

Each General\_rib has a walls\_top\_floor defined by one or more Planar\_element. Each Planar\_element defines the walls\_top\_floor for zero, one, or many General\_rib objects.

NOTE This assertion is established through the Shape\_select.

#### **4.3.200 General\_rib to Face\_shape\_element**

Each General\_rib has a walls\_top\_floor defined by one or more Face\_shape\_element. Each Face\_shape\_element defines the walls\_top\_floor for zero, one, or many General\_rib objects.

NOTE This assertion is established through the Shape\_select.

#### **4.3.201 Geometric\_tolerance to Material\_condition\_modifier**

Each Geometric\_tolerance has the modifier\_control defined by zero or one Material\_condition\_modifier. Each Material\_condition\_modifier defines the modifier\_control for zero, one, or many Geometric\_tolerance objects.

#### **4.3.202 Geometric\_tolerance to Shape\_aspect**

Each Geometric\_tolerance has the applied\_shape defined by one or more Shape\_aspect objects. Each Shape\_aspect defines the applied\_shape for zero, one, or many Geometric\_tolerance objects.

#### **4.3.203 Geometric\_tolerance to Tolerance\_zone**

Each Geometric\_tolerance has the zone\_definition defined by zero or one Tolerance\_zone. Each Tolerance\_zone defines the zone\_definition for zero, one, or many Geometric\_tolerance objects.

#### **4.3.204 Geometric\_tolerance\_precedence\_relationship to Geometric\_tolerance**

Each Geometric\_tolerance\_precedence\_relationship has the base\_shape\_tolerance defined by exactly one Geometric\_tolerance. Each Geometric\_tolerance defines the base\_shape\_tolerance for zero, one, or many Geometric\_tolerance\_precedence\_relationship objects.

Each Geometric\_tolerance\_precedence\_relationship has the pattern\_shape\_tolerance defined by exactly one Geometric\_tolerance. Each Geometric\_tolerance defines the pattern\_shape\_tolerance for zero, one, or many Geometric\_tolerance\_precedence\_relationship objects.

#### **4.3.205 Geometry\_characteristic to Direction\_element**

Each Geometry\_characteristic has an orientation defined by exactly one Direction\_element. Each Direction\_element defines the orientation for zero, one, or many Geometry\_characteristic objects.

#### **4.3.206 Geometry\_characteristic to Numeric\_range**

Each Geometry\_characteristic has a character\_size defined by zero or one Numeric\_range. Each Numeric\_range defines the character\_size for zero, one, or many Geometry\_characteristic objects.

#### **4.3.207 Geometry\_characteristic to Shape\_element**

Each Geometry\_characteristic has a shape defined by zero or one Shape\_element. Each Shape\_element defines the shape for zero, one, or many Geometry\_characteristic objects.

#### **4.3.208 Heat to Date**

Each Heat has a production\_date defined by exactly one Date. Each Date defines the production\_date for zero, one, or many Heat objects.

#### **4.3.209 Heat\_treat to Heat\_treat\_requirement**

Each Heat\_treat has a requirement defined by exactly one Heat\_treat\_requirement. Each Heat\_treat\_requirement defines the requirement for zero, one, or many Heat\_treat objects.

#### **4.3.210 Heat\_treat\_requirement to Mechanical\_property\_requirement**

Each Heat\_treat\_requirement has a mechanical\_property defined by one or more Mechanical\_property\_requirement. Each Mechanical\_property\_requirement defines the mechanical\_property for zero, one, or many Heat\_treat\_requirement objects.

#### **4.3.211 Heat\_treat\_requirement to Specification**

Each Heat\_treat\_requirement has a heat\_treat\_specification defined by zero or one Specification. Each Specification defines the heat\_treat\_specification for zero, one, or many Heat\_treat\_requirement objects.

#### **4.3.212 Heat\_treat\_requirement to Process\_property**

Each Heat\_treat\_requirement has a heat\_treatment\_per\_design\_data defined by exactly one Process\_property. Each Process\_property defines the heat\_treatment\_per\_design\_data for zero, one, or many Heat\_treat\_requirement objects.

#### **4.3.213 Height\_dimension to Shape\_aspect**

Each Height\_dimension has a path defined by exactly one Shape\_aspect. Each Shape\_aspect defines the path for zero, one, or many Height\_dimension objects.

#### **4.3.214 Illustration to View\_reference**

Each Illustration has the is\_owned\_by defined by zero, one or many View\_reference objects. Each View\_reference defines the is\_owned\_by for zero, one or many Illustration objects.

#### **4.3.215 Ingate\_contacts to Production\_pattern\_definition**

Each Ingate\_contacts has the applied\_to defined by exactly one Production\_pattern\_definition. Each Production\_pattern\_definition defines the applied\_to for zero, one, or many Ingate\_contacts objects.

#### **4.3.216 Ingate\_contacts to Shape\_element**

Each Ingate\_contacts has the break\_off\_connection defined by zero or one Shape\_element. Each Shape\_element defines the break\_off\_connection for zero, one, or many Ingate\_contacts objects.

#### **4.3.217 Ingate\_contacts to Connection\_transition**

Each Ingate\_contacts has the runner\_transition defined by exactly one Connection\_transition objects. Each Connection\_transition defines the runner\_transition for zero, one, or many Ingate\_contacts objects.

Each Ingate\_contacts has the pattern\_transition defined by exactly one Connection\_transition objects. Each Connection\_transition defines the pattern\_transition for zero, one, or many Ingate\_contacts objects.

#### **4.3.218 Ingate\_contacts to Runner**

Each Ingate\_contacts has the runner\_connection defined by exactly one Runners. Each Runners defines the runner\_connection for zero, one, or many Ingate\_contacts objects.

#### **4.3.219 Ingate to Ingate\_contacts**

Each Ingate has the contacts defined by exactly one Ingate\_contacts. Each Ingate\_contacts defines the contacts for zero, one, or many Ingate objects.

#### **4.3.220 Ingate to Item\_size**

Each Ingate has the ingate\_shape defined by zero or one Item\_size. Each Item\_size defines the ingate\_shape for zero, one, or many Ingate objects.

#### **4.3.221 Inspection\_activity to Approval**

Each Inspection\_activity has the foundry\_approval defined by exactly one Approval. Each Approval defines the foundry\_approval for zero, one, or many Inspection\_activity objects.

#### **4.3.222 Inspection\_activity to Casting\_verification**

Each Inspection\_activity has the quality\_assurance defined by exactly one Casting\_verification. Each Casting\_verification defines the quality\_assurance for zero, one, or many Inspection\_activity objects.

#### **4.3.223 Inspection\_activity to Inspection\_plan**

Each `Inspection_activity` has the `customer_requirements` defined by one or more `Inspection_plan`. Each `Inspection_plan` defines the `customer_requirements` for zero, one, or many `Inspection_activity` objects.

#### **4.3.224 Inspection\_activity to Part\_version**

Each `Inspection_activity` has the `inspected_part` defined by exactly one `Part_version`. Each `Part_version` defines the `inspected_part` for zero, one, or many `Inspection_activity` objects.

#### **4.3.225 Inspection\_activity to Product\_quality\_report**

Each `Inspection_activity` has the `inspection_report` defined by exactly one `Product_quality_report`. Each `Product_quality_report` defines the `inspection_report` for zero, one, or many `Inspection_activity` objects.

#### **4.3.226 Inspection\_or\_test\_requirement to Customer\_casting\_requirement**

Each `Inspection_or_test_requirement` has the `inspection_against` defined by exactly one `Customer_casting_requirement`. Each `Customer_casting_requirement` defines the `inspection_against` for zero, one, or many `Inspection_or_test_requirement` objects.

#### **4.3.227 Inspection\_or\_test\_requirement to Numeric\_parameter**

Each `Inspection_or_test_requirement` has the `frequency` defined by exactly one `Numeric_parameter`. Each `Numeric_parameter` defines the `frequency` for zero, one, or many `Inspection_or_test_requirement` objects.

Each `Inspection_or_test_requirement` has the `sampling_run_size` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `sampling_run_size` for zero, one, or many `Inspection_or_test_requirement` objects.

#### **4.3.228 Inspection\_or\_test\_requirement to Descriptive\_parameter**

Each `Inspection_or_test_requirement` has the `test_name` defined by zero or one `Descriptive_parameter`. Each `Descriptive_parameter` defines the `test_name` for zero, one, or many `Inspection_or_test_requirement` objects.

#### **4.3.229 Inspection\_or\_test\_requirement to Specification**

Each `Inspection_or_test_requirement` has the `production_quality_requirement_standard` defined by one or more `Specification`. Each `Specification` defines the `production_quality_requirement_standard` for zero, one, or many `Inspection_or_test_requirement` objects.

#### **4.3.230 Inspection\_or\_test\_result to Sampled\_set**

Each `Inspection_or_test_result` has the `belongs_to` defined by exactly one `Sampled_set`. Each `Sampled_set` defines the `belongs_to` for zero, one, or many `Inspection_or_test_result` objects.

#### **4.3.231 Inspection\_or\_test\_result to Checking\_aid\_tool**

Each `Inspection_or_test_result` has the `aid_used` defined by zero, one, or many `Checking_aid_tool`. Each `Checking_aid_tool` defines the `aid_used` for zero, one, or many `Inspection_or_test_result` objects.

#### **4.3.232 Inspection\_or\_test\_result to Shape\_aspect**

Each `Inspection_or_test_result` has the `applies_to` defined by zero or one `Shape_aspect`. Each `Shape_aspect` defines the `applies_to` for zero, one, or many `Inspection_or_test_result` objects.

#### **4.3.233 Inspection\_plan to Inspection\_or\_test\_requirement**

Each `Inspection_plan` has the `customer_requirements` defined by one or more `Inspection_or_test_requirement`. Each `Inspection_or_test_requirement` defines the `customer_requirements` for zero, one, or many `Inspection_plan` objects.

NOTE This assertion is established through `customer_requirement_select`.

#### **4.3.234 Inspection\_plan to Special\_inspection\_requirement**

Each `Inspection_plan` has the `customer_requirements` defined by one or more `Special_inspection_requirement`. Each `Special_inspection_requirement` defines the `customer_requirements` for zero, one, or many `Inspection_plan` objects.

NOTE This assertion is established through `customer_requirement_select`.

#### **4.3.235 Integration\_interval to Numeric\_parameter**

Each `Integration_interval` has the `delta_time` defined by exactly one `Numeric_parameter`. Each `Numeric_parameter` defines the `delta_time` for zero, one, or many `Integration_interval` objects.

Each `Integration_interval` has the `length_of_elapsed_time_interval` defined by exactly one `Numeric_parameter`. Each `Numeric_parameter` defines the `length_of_elapsed_time_interval` for zero, one, or many `Integration_interval` objects.

#### **4.3.236 Investment\_casting\_tooling to Cast\_part\_with\_rigging**

Each `Investment_casting_tooling` has the `investment_tooling_or_prototype` defined by one or more `Cast_part_with_rigging`. Each `Cast_part_with_rigging` defines the `investment_tooling_or_prototype` for zero, one, or many `Investment_casting_tooling` objects.

NOTE This assertion is established through `prototypeing_or_mould_select`.

#### **4.3.237 Investment\_casting\_tooling to Investment\_casting\_tooling\_definition**

Each `Investment_casting_tooling` has the `investment_tooling_or_prototype` defined by one or more `Investment_casting_tooling_definition`. Each `Investment_casting_tooling_definition` defines the `investment_tooling_or_prototype` for zero, one, or many `Investment_casting_tooling` objects.

NOTE This assertion is established through `prototypeing_or_mould_select`.

#### **4.3.238 Investment\_casting\_tooling\_definition to Production\_investment\_-cast\_mould**

Each `Investment_casting_tooling_definition` has the produces by one or more `Production_investment_-cast_mould`. Each `Production_investment_cast_mould` defines the produces for zero, one, or many `Investment_casting_tooling_definition` objects.

NOTE This assertion is established through `wax_or_eps_select`.

#### **4.3.239 Investment\_casting\_tooling\_definition to Lost\_foam\_casting\_die**

Each `Investment_casting_tooling_definition` has the produces by one or more `Lost_foam_casting_die`. Each `Lost_foam_casting_die` defines the produces for zero, one, or many `Investment_casting_tooling_definition` objects.

NOTE This assertion is established through `wax_or_eps_select`.

#### **4.3.240 Investment\_casting\_tooling\_definition to Production\_core\_box**

Each `Investment_casting_tooling_definition` has the `core_tooling` defined by zero, one, or many `Production_core_box`. Each `Production_core_box` defines the `core_tooling` for zero, one, or many `Investment_casting_tooling_definition` objects.

#### **4.3.241 Investment\_casting\_tooling\_definition to Assembly**

Each `Investment_casting_tooling_definition` has the `pattern_section_assembly` defined by zero, one, or many `Assembly`. Each `Assembly` defines the `pattern_section_assembly` for zero, one, or many `Investment_casting_tooling_definition` objects.



#### **4.3.242 Investment\_casting\_tooling\_definition to Shape\_element**

Each `Investment_casting_tooling_definition` has the `pattern_section_definition` defined by one or more `Shape_element`. Each `Shape_element` defines the `pattern_section_definition` for zero, one, or many `Investment_casting_tooling_definition` objects.

#### **4.3.243 Investment\_casting\_tooling\_definition to Investment\_mould**

Each `Investment_casting_tooling_definition` has the `tree_structure` defined by zero or one `Investment_mould`. Each `Investment_mould` defines the `tree_structure` for zero, one, or many `Investment_casting_tooling_definition` objects.

#### **4.3.244 Investment\_casting\_tooling\_definition to Sprue\_and\_runner\_mould**

Each `Investment_casting_tooling_definition` has the `metal_flow_system` defined by exactly one `Sprue_and_runner_mould`. Each `Sprue_and_runner_mould` defines the `metal_flow_system` for zero, one, or many `Investment_casting_tooling_definition` objects.

#### **4.3.245 Investment\_design\_feature to Orientation**

Each `Investment_design_feature` has the placement defined by exactly one `Orientation`. Each `Orientation` defines the placement for zero, one, or many `Investment_design_feature` objects.

#### **4.3.246 Item\_size to Numeric\_parameter**

Each `Item_size` has the `item_height` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `item_height` for zero, one, or many `Item_size` objects.

Each `Item_size` has the `item_length` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `item_length` for zero, one, or many `Item_size` objects.

Each `Item_size` has the `item_width` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `item_width` for zero, one, or many `Item_size` objects.

Each `Item_size` has the `item_diameter` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `item_diameter` for zero, one, or many `Item_size` objects.

Each `Item_size` has the `item_draft` defined by zero or one `Numeric_parameter`. Each `Numeric_parameter` defines the `item_draft` for zero, one, or many `Item_size` objects.

#### **4.3.247 Length\_dimension to Shape\_aspect**

Each `Length_dimension` has a path defined by exactly one `Shape_aspect`. Each `Shape_aspect` defines the path for zero, one, or many `Length_dimension` objects.

#### **4.3.248 Library\_part\_assignment to Property\_value**

Each Library\_part\_assignment has definitional\_property\_value\_pairs defined by zero, one, or many Property\_value. Each Property\_value defines the definitional\_property\_value\_pairs for zero, one, or many Library\_part\_assignment objects.

#### **4.3.249 Library\_part\_assignment to Class\_BSU**

Each Library\_part\_assignment has definitional\_class\_BSU defined by exactly one Class\_BSU. Each Class\_BSU defines the definitional\_class\_BSU for zero, one, or many Library\_part\_assignment objects

#### **4.3.250 Location\_dimension\_tolerance to Orientation**

Each Location\_dimension\_tolerance has the plane and direction defined by zero or one Shape\_aspect. Each Orientation defines the plane and direction for zero, one, or many Location\_dimension\_tolerance objects.

#### **4.3.251 Location\_tolerance to Shape\_element**

Each Location\_tolerance has the termination\_shape defined by exactly one Shape\_element. Each Shape\_element defines the termination\_shape for zero, one, or many Location\_dimension objects.

Each Location\_dimension has the origin\_shape defined by exactly one Shape\_element. Each Shape\_element defines the origin\_shape for zero, one, or many Location\_dimension objects.

#### **4.3.252 Loose\_piece to Shape\_element**

Each Loose\_piece has the shape defined by one or more Shape\_element. Each Shape\_element defines the shape for zero, one, or many Loose\_piece objects.

#### **4.3.253 Lost\_foam\_casting\_die to Cooling\_port**

Each Lost\_foam\_casting\_die has the mould\_cooling defined by one or more Cooling\_port. Each Cooling\_port defines the mould\_cooling for zero, one, or many Lost\_foam\_casting\_die objects.

#### **4.3.254 Lost\_foam\_casting\_die to Die\_clamping**

Each Lost\_foam\_casting\_die has the stationary\_die defined by exactly one Die\_clamping. Each Die\_clamping defines the stationary\_die for zero, one, or many Lost\_foam\_casting\_die objects.

Each Lost\_foam\_casting\_die has the movable\_die\_item defined by one or more Die\_clamping. Each Die\_clamping defines the stationary\_die for zero, one, or many Lost\_foam\_casting\_die objects.

NOTE This assertion is established through the movable\_slide\_select.

#### **4.3.255 Lost\_foam\_casting\_die to Slide**

Each `Lost_foam_casting_die` has the `movable_die_item` defined by one or more `Slide`. Each `Slide` defines the `stationary_die` for zero, one, or many `Lost_foam_casting_die` objects.

NOTE This assertion is established through the `movable_slide_select`.

#### **4.3.256 Lost\_foam\_casting\_die to Flask**

Each `Lost_foam_casting_die` has the `defines_flask_type` defined by exactly one `Flask`. Each `Flask` defines the `defines_flask_type` for zero, one, or many `Lost_foam_casting_die` objects.

#### **4.3.257 Lost\_foam\_casting\_die to Machine\_mounting**

Each `Lost_foam_casting_die` has the `stationary_die_mounting` defined by exactly one `Machine_mounting`. Each `Machine_mounting` defines the `stationary_die_mounting` for zero, one, or many `Lost_foam_casting_die` objects.

#### **4.3.258 Lost\_foam\_casting\_die to Item\_size**

Each `Lost_foam_casting_die` has the `fill_gun_size` defined by one or more `Item_size`. Each `Item_size` defines the `fill_gun_size` for zero, one, or many `Lost_foam_casting_die` objects.

#### **4.3.259 Lost\_foam\_casting\_die to Gating\_design\_feature**

Each `Lost_foam_casting_die` has the `gating_requirements` defined by one or more `Gating_design_feature`. Each `Gating_design_feature` defines the `gating_requirements` for zero, one, or many `Lost_foam_casting_die` objects.

#### **4.3.260 Lost\_foam\_casting\_die to Planar\_element**

Each `Lost_foam_casting_die` has the `fill_gun_location` defined by one or more `Planar_element`. Each `Planar_element` defines the `fill_gun_location` for zero, one, or many `Lost_foam_casting_die` objects.

Each `Lost_foam_casting_die` has the `steam_line_location` defined by one or more `Planar_element`. Each `Planar_element` defines the `steam_line_location` for zero, one, or many `Lost_foam_casting_die` objects.

#### **4.3.261 Lost\_foam\_casting\_die to In\_facility\_location**

Each `Lost_foam_casting_die` has the `tool_location` defined by two or more `In_facility_location`. Each `In_facility_location` defines the `tool_location` for zero, one, or many `Lost_foam_casting_die` objects.

NOTE This assertion is established through the `Tool_location_select_select`.

#### **4.3.262 Lost\_foam\_casting\_die to Person\_in\_organization**

Each Lost\_foam\_casting\_die has the tool\_location defined by two or more Person\_in\_organization. Each Person\_in\_organization the tool\_location for zero, one, or many Lost\_foam\_casting\_die objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### **4.3.263 Lot to date**

Each Lot has the production\_date defined by exactly one Date. Each Date defines the production\_date for zero, one, or many Lot objects.

#### **4.3.264 Machine\_mounting to Equipment\_size**

Each Machine\_mounting has the dimensions defined by exactly one Equipment\_size. Each Equipment\_size defines the dimensions for zero, one, or many Machine\_mounting objects.

#### **4.3.265 Machine\_mounting to Location\_element**

Each Machine\_mounting defines the location by exactly one Location\_element. Each Location\_element defines the location by for zero, one, or many Machine\_mounting objects.

#### **4.3.266 Machining\_activity to External\_schema\_definition**

Each Machining\_activity has the iso\_10303\_240 defined by zero or one External\_schema\_definition. Each External\_schema\_definition defines the iso\_10303\_240 for zero, one, or many Machining\_activity objects.

#### **4.3.267 Machining\_allowance to Manufacturing\_feature**

Each Machining\_allowance has the associated\_feature defined by exactly one Manufacturing\_feature. Each Manufacturing\_feature defines the associated\_feature for zero, one, or many Machining\_allowance objects.

#### **4.3.268 Machining\_allowance to Numeric\_parameter**

Each Machining\_allowance has the allowance\_value defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the allowance\_value for zero, one, or many Machining\_allowance objects.

#### **4.3.269 Machining\_feature to Orientation**

Each Machining\_feature has placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Machining\_feature objects.

#### **4.3.270 Manufacturing\_feature\_group to Manufacturing\_feature**

Each Manufacturing\_feature\_group has the feature\_groups defined by two or more Manufacturing\_feature objects. Each Manufacturing\_feature defines the feature\_groups for zero, one, or many Manufacturing\_feature\_group objects.

NOTE This assertion is established through Manufacturing\_group\_select.

#### **4.3.271 Manufacturing\_feature\_group to Manufacturing\_feature\_group**

Each Manufacturing\_feature\_group has the feature\_groups defined by two or more Manufacturing\_feature\_group objects. Each Manufacturing\_feature\_group defines the feature\_groups for zero, one, or many Manufacturing\_feature\_group objects.

NOTE This assertion is established through Manufacturing\_group\_select.

#### **4.3.272 Manufacturing\_process to Setup\_activity**

Each Manufacturing\_process has the setup defined by one or more Setup\_activity objects. Each Setup\_activity defines the setup for zero, one, or many Manufacturing\_process objects.

#### **4.3.273 Manufacturing\_process to Casting\_activity**

Each Manufacturing\_process has the assigned\_operation defined by one or more Casting\_activity objects. Each Casting\_activity defines the assigned\_operation for zero, one, or many Manufacturing\_process objects.

#### **4.3.274 Master\_pattern to Machining\_allowance**

Each Master\_pattern has the machining\_allowance\_on\_master\_pattern defined by one or more Machining\_allowance. Each Machining\_allowance defines the machining\_allowance\_on\_master\_pattern for zero, one, or many Master\_pattern objects.

#### **4.3.275 Master\_sample to Heat**

Each Master\_sample has the belongs\_to\_heat defined by zero or one Heat objects. Each Heat defines the belongs\_to\_heat for zero, one or many Master\_sample objects.

#### **4.3.276 Master\_sample to Lot**

Each Master\_sample has the belongs\_to\_lot defined by zero or one Lot objects. Each Lot defines the belongs\_to\_lot for zero, one or many Master\_sample objects.

#### **4.3.277 Master\_sample to Date**

Each Master\_sample has the production\_date defined by exactly one Date objects. Each Date defines the production\_date for zero, one or many Master\_sample objects.

#### **4.3.278 Master\_sample to Design\_part**

Each Master\_sample has the product\_specification\_used defined by exactly one Design\_part objects. Each Design\_part defines the product\_specification\_used for zero, one or many Master\_sample objects.

#### **4.3.279 Master\_sample to External\_schema\_definition**

Each Master\_sample has the process\_plan\_used defined by exactly one External\_schema\_definition objects. Each External\_schema\_definition defines the process\_plan\_used for zero, one or many Master\_sample objects.

NOTE This assertion is established through process\_plan\_select.

#### **4.3.280 Master\_sample to Process\_plan**

Each Master\_sample has the process\_plan\_used defined by exactly one Process\_plan objects. Each Process\_plan defines the process\_plan\_used for zero, one or many Master\_sample objects.

NOTE This assertion is established through process\_plan\_select.

#### **4.3.281 Material to Material\_composition**

Each Material has the composition defined by zero, one, or many Material\_composition objects. Each Material\_composition defines the composition for zero, one, or many Material objects.

#### **4.3.282 Material to Material\_property**

Each Material has the material\_characteristics defined by one or more Material\_property objects. Each Material\_property defines the material\_characteristics for zero, one, or many Material objects.

#### **4.3.283 Material to Material\_structure**

Each Material has the structure defined by zero, one, or many Material\_structure objects. Each Material\_structure defines the structure for zero, one, or many Material objects.

#### **4.3.284 Material to Specification**

Each Material has the material\_specification defined by zero, one, or many Specification objects. Each Specification defines the material\_characteristics for zero, one, or many Material objects.

#### **4.3.285 Material\_composition to Composition\_element**

Each Material\_composition has the elements defined by one or more Composition\_element objects. Each Composition\_element defines the elements for zero, one, or many Material\_composition objects.

#### **4.3.286 Material\_property to Hardness**

Each Material\_property has the material\_hardness defined by exactly one Hardness objects. Each Hardness defines the material\_hardness for zero, one, or many Material\_property objects.

#### **4.3.287 Material\_property to Numeric\_parameter**

Each Material\_property has the melt\_temperature defined by exactly one Numeric\_parameters objects. Each Numeric\_parameter defines the melt\_temperature for zero, one, or many Material\_property objects.

#### **4.3.288 Material\_property to Property\_parameter**

Each Material\_property has the property\_characteristics defined by exactly one Property\_parameters objects. Each Property\_parameter defines the property\_characteristics for zero, one, or many Material\_property objects.

#### **4.3.289 Material\_structure to Illustration**

Each Material\_structure has the illustration defined by exactly one Illustration objects. Each Illustration defines the illustration for zero, one, or many Material\_structure objects.

#### **4.3.290 Material\_structure to Material\_structure\_element**

Each Material\_structure has the element defined by one or more Material\_structure\_element objects. Each Material\_structure\_element defines the element for zero, one, or many Material\_structure objects.

#### **4.3.291 Material\_structure to Specification**

Each Material\_structure has the reference\_to defined by exactly one Specification objects. Each Specification defines the reference\_to for zero, one, or many Material\_structure objects.

#### **4.3.292 Material\_structure\_element to Geometry\_characteristic**

Each Material\_structure\_element has the characteristic\_geometry defined by exactly one Geometry\_characteristic objects. Each Geometry\_characteristic defines the characteristic\_geometry for zero, one, or many Material\_structure\_element objects.

#### **4.3.293 Material\_structure\_element to Material**

Each Material\_structure\_element has the made\_of defined by exactly one Material objects. Each Material defines the made\_of for zero, one, or many Material\_structure\_element objects.

#### **4.3.294 Material\_structure\_element to Numeric\_range**

Each Material\_structure\_element has the amount defined by exactly one Numeric\_range objects. Each Numeric\_range defines the amount for zero, one, or many Material\_structure\_element objects.

#### **4.3.295 Material\_usage to Descriptive\_parameter**

Each Material\_usage has the amount\_used defined by exactly one Descriptive\_parameter objects. Each Descriptive\_parameter defines the amount\_used for zero, one, or many Material\_usage objects.

#### **4.3.296 Material\_usage to Material**

Each Material\_usage has the actual\_material defined by exactly one Material objects. Each Material defines the actual\_material for zero, one, or many Material\_usage objects.

#### **4.3.297 Material\_usage\_record to Numeric\_parameter**

Each Material\_usage\_record has the amount defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the amount for zero, one, or many Material\_usage\_record objects.

#### **4.3.298 Material\_usage\_record to Material\_usage**

Each Material\_usage\_record has the recorded\_usage defined by exactly one Material\_usage objects. Each Material\_usage defines the recorded\_usage for zero, one, or many Material\_usage\_record objects.

#### **4.3.299 Mechanical\_property\_requirements to Hardness**

Each Mechanical\_property\_requirements has the material\_hardness defined by exactly one Hardness objects. Each Hardness defines the material\_hardness for zero, one, or many Mechanical\_property\_requirements objects.

#### **4.3.300 Mechanical\_property\_requirements to Material\_property**

Each Mechanical\_property\_requirements has the tensile\_strength defined by exactly one Material\_property objects. Each Material\_property defines the tensile\_strength for zero, one, or many Mechanical\_property\_requirements objects.

Each Mechanical\_property\_requirements has the yeild\_strength defined by zero or one Material\_property objects. Each Material\_property defines the yeild\_strength for zero, one, or many Mechanical\_property\_requirements objects.



#### **4.3.301 Mechanical\_property\_requirements to Specification**

Each Mechanical\_property\_requirements has the requirement\_specification defined by exactly one Specification objects. Each Specification defines the requirement\_specification for zero, one, or many Mechanical\_property\_requirements objects.

#### **4.3.302 Mechanical\_stress to Descriptive\_parameter**

Each Mechanical\_stress has the type\_of\_stress defined by zero, one, or many Descriptive\_parameter objects. Each Descriptive\_parameter defines the type\_of\_stress for zero, one, or many Mechanical\_stress objects.

#### **4.3.303 Mechanical\_stress to Numeric\_parameter**

Each Mechanical\_stress has the maximum\_psi\_design\_stress defined by zero or one Numeric\_parameter objects. Each Numeric\_parameter defines the maximum\_psi\_design\_stress for zero, one, or many Mechanical\_stress objects.

Each Mechanical\_stress has the percent\_safety\_factor\_required defined by zero or one Numeric\_parameter objects. Each Numeric\_parameter defines the percent\_safety\_factor\_required for zero, one, or many Mechanical\_stress objects.

#### **4.3.304 Meshing\_condition to Numeric\_range**

Each Meshing\_condition has the density\_of\_elements defined by zero or one Numeric\_range objects. Each Numeric\_range defines the density\_of\_elements for zero, one, or many Meshing\_condition objects.

Each Meshing\_condition has the density\_of\_nodes defined by zero or one Numeric\_range objects. Each Numeric\_range defines the density\_of\_nodes for zero, one, or many Meshing\_condition objects.

#### **4.3.305 Meshing\_condition to Shape\_aspect**

Each Meshing\_condition has the area\_to\_be\_meshed defined by zero, one, or many Shape\_aspect objects. Each Shape\_aspect defines the area\_to\_be\_meshed for zero, one, or many Meshing\_condition objects.

#### **4.3.306 Metalcaster\_simulation to Simulation\_tooling\_report**

Each Metalcaster\_simulation has the suggestion\_for\_changes defined by zero or one Simulation\_tooling\_report objects. Each Simulation\_tooling\_report defines the suggestion\_for\_changes for zero, one, or many Metalcaster\_simulation objects.

#### **4.3.307 Metalcaster\_simulation to Simulation\_exception\_report**

Each Metalcaster\_simulation has the defect\_results defined by zero, one, or many Simulation\_exception\_report objects. Each Simulation\_exception\_report defines the defect\_results for zero, one, or many Metalcaster\_simulation objects.

#### **4.3.308 Metalcaster\_simulation to Gating\_system**

Each Metalcaster\_simulation has the proposed\_gating defined by exactly one Gating\_system objects. Each Gating\_system defines the proposed\_gating for zero, one, or many Metalcaster\_simulation objects.

#### **4.3.309 Metalcaster\_simulation to Pattern\_equipment\_available**

Each Metalcaster\_simulation has the available\_tooling defined by zero or one Pattern\_equipment\_available objects. Each Pattern\_equipment\_available defines the available\_tooling for zero, one, or many Metalcaster\_simulation objects.

#### **4.3.310 Mismatch\_tolerance to Numeric\_parameter**

Each Mismatch\_tolerance has the maximum\_amount defined by exactly one Numeric\_parameter objects. Each Numeric\_parameter defines the maximum\_amount for zero, one, or many Mismatch\_tolerance objects.

#### **4.3.311 Mismatch\_tolerance to Parting\_surface**

Each Mismatch\_tolerance has the with\_respect\_to defined by exactly one Parting\_surface objects. Each Parting\_surface defines the with\_respect\_to for zero, one, or many Mismatch\_tolerance objects.

#### **4.3.312 Model\_element to Geometric\_model**

Each Model\_element defines the element by exactly one Geometric\_model. Each Geometric\_model defines the element for zero, one, or many Model\_element objects.

#### **4.3.313 Model\_element to Shape\_representation\_type**

Each Model\_element defines the representation\_type by exactly one Shape\_representation\_type. Each Shape\_representation\_type defines the representation\_type for zero, one, or many Model\_element objects.

#### **4.3.314 Mould\_box to Item\_size**

Each Mould\_box defines the box\_size exactly one Item\_size. Each Item\_size defines the box\_size for zero, one, or many Mould\_box objects.

**4.3.315 Mould\_box to Descriptive\_parameter**

Each Mould\_box defines the description by exactly one Descriptive\_parameter. Each Descriptive\_parameter defines the description by for zero, one, or many Mould\_box objects.

**4.3.316 Mould\_locks to Item\_size**

Each Mould\_locks defines the lock\_dimensions\_by exactly one Item\_size. Each Item\_size defines the lock\_dimensions\_by for zero, one, or many Mould\_locks objects.

**4.3.317 Mould\_locks to Descriptive\_parameter**

Each Mould\_locks defines the description by exactly one Descriptive\_parameter. Each Descriptive\_parameter defines the description by for zero, one, or many Mould\_locks objects.

**4.3.318 Non\_casting\_operations to Cast\_part**

Each Non\_casting\_operations has the part\_geometry\_with\_machine\_allowance defined by zero or one Cast\_part. Each Cast\_part defines the part\_geometry\_with\_machine\_allowance for zero, one, or many Non\_casting\_operations objects.

**4.3.319 Non\_casting\_operations to Design\_part**

Each Non\_casting\_operations has the part\_geometry\_with\_no\_machine\_allowance defined by zero or one Design\_part. Each Design\_part defines the part\_geometry\_with\_no\_machine\_allowance for zero, one, or many Non\_casting\_operations objects.

**4.3.320 Non\_casting\_operations to Descriptive\_parameter**

Each Non\_casting\_operations has the operation\_description defined by zero or one Descriptive\_parameter. Each Descriptive\_parameter defines the operation\_description for zero, one, or many Non\_casting\_operations objects.

**4.3.321 Non\_casting\_operations to Finishing\_or\_machining\_operations**

Each Non\_casting\_operations has the customer\_requirements defined by zero or one Finishing\_or\_machining\_operations. Each Finishing\_or\_machining\_operations defines the customer\_requirements for zero, one, or many Non\_casting\_operations objects.

**4.3.322 Non\_permanent\_moulding\_process to Flask**

Each Non\_permanent\_moulding\_process has flask\_type defined by exactly one Flask. Each Flask defines the flask\_type for zero, one, or many Non\_permanent\_moulding\_process objects.

NOTE This assertion is established through the flask\_select.

#### **4.3.323 Non\_permanent\_moulding\_process to Flaskless**

Each Non\_permanent\_moulding\_process has flask\_type defined by exactly one Flaskless. Each Flaskless defines the flask\_type for zero, one, or many Non\_permanent\_moulding\_process objects.

NOTE This assertion is established through the flask\_select.

#### **4.3.324 Non\_permanent\_moulding\_process to Investment\_mould**

Each Non\_permanent\_moulding\_process has pattern\_tooling defined by exactly one Investment\_mould. Each Investment\_mould defines the pattern\_tooling for zero, one, or many Non\_permanent\_moulding\_process objects.

NOTE This assertion is established through the pattern\_tooling\_select.

#### **4.3.325 Non\_permanent\_moulding\_process to Lost\_foam\_casting\_die**

Each Non\_permanent\_moulding\_process has pattern\_tooling defined by exactly one Lost\_foam\_casting\_die. Each Lost\_foam\_casting\_die defines the pattern\_tooling for zero, one, or many Non\_permanent\_moulding\_process objects.

NOTE This assertion is established through the pattern\_tooling\_select.

#### **4.3.326 Non\_permanent\_moulding\_process to Production\_pattern\_definition**

Each Non\_permanent\_moulding\_process has pattern\_tooling defined by exactly one Production\_pattern\_definition. Each Production\_pattern\_definition defines the pattern\_tooling for zero, one, or many Non\_permanent\_moulding\_process objects.

NOTE This assertion is established through the pattern\_tooling\_select.

#### **4.3.327 Numeric\_parameter\_with\_tolerance to Plus\_minus\_value**

Each Numeric\_parameter\_with\_tolerance has the implicit\_tolerance defined by exactly one Plus\_minus\_value. Each Plus\_minus\_value defines the implicit\_tolerance for zero, one, or many Numeric\_parameter\_with\_tolerance objects.

NOTE This assertion is established through the Numeric\_parameter\_tolerance\_select.

#### **4.3.328 Numeric\_parameter\_with\_tolerance to Tolerance\_limit**

Each Numeric\_parameter\_with\_tolerance has the implicit\_tolerance defined by exactly one Tolerance\_limit. Each Tolerance\_limit defines the implicit\_tolerance for zero, one, or many Numeric\_parameter\_with\_tolerance objects.

NOTE This assertion is established through the Numeric\_parameter\_tolerance\_select.

#### **4.3.329 Numeric\_parameter\_with\_tolerance to Tolerance\_range**

Each `Numeric_parameter_with_tolerance` has the `implicit_tolerance` defined by exactly one `Tolerance_range`. Each `Tolerance_range` defines the `implicit_tolerance` for zero, one, or many `Numeric_parameter_with_tolerance` objects.

NOTE This assertion is established through the `Numeric_parameter_tolerance_select`.

#### **4.3.330 Numeric\_parameter\_with\_tolerance to Limits\_and\_fits**

Each `Numeric_parameter_with_tolerance` has the `implicit_tolerance` defined by exactly one `Limits_and_fits`. Each `Limits_and_fits` defines the `implicit_tolerance` for zero, one, or many `Numeric_parameter_with_tolerance` objects.

NOTE This assertion is established through the `Numeric_parameter_tolerance_select`.

#### **4.3.331 Numeric\_range to Tolerance\_range**

Each `Numeric_range` has the `value_range` defined by exactly one `Tolerance_range`. Each `Tolerance_range` defines the `value_range` for zero, one, or many `Numeric_range` objects.

#### **4.3.332 Object\_element\_shape\_representation to Geometric\_model**

Each `Object_element_shape_representation` has the `shape_definition` defined by exactly one `Geometric_model`. Each `Geometric_model` defines the `shape_definition` for zero, one, or many `Object_element_shape_representation` objects.

#### **4.3.333 Object\_element\_shape\_representation to Shape\_representation\_type**

Each `Object_element_shape_representation` has the `representation_type` defined by exactly one `Shape_representation_type`. Each `Shape_representation_type` defines the `representation_type` for zero, one, or many `Object_element_shape_representation` objects.

#### **4.3.334 Organization to Address**

Each `Organization` has the `address` defined by exactly one `Address`. Each `Address` defines the `address` for zero, one, or many `Organization` objects.

#### **4.3.335 Other\_inspection to Inspection\_or\_test\_requirement**

Each `Other_inspection` has the `requirement` defined by one or more `Inspection_or_test_requirement`. Each `Inspection_or_test_requirement` defines the `requirement` for zero, one, or many `Other_inspection` objects.

#### **4.3.336 Part\_dimensioning\_standard to Part\_version**

Each Part\_dimensioning\_standard has the applied\_part defined by exactly one Part\_version. Each Part\_version defines the applied\_part for zero, one, or many Part\_dimensioning\_standard objects.

#### **4.3.337 Part\_property to Property\_parameter**

Each Part\_property has the property\_characteristic defined by zero, one, or many Property\_parameter objects. Each Property\_parameter defines the property\_characteristic for exactly one Part\_property.

#### **4.3.338 Part\_relationship to Cast\_part**

Each Part\_relationship has the to\_be\_cast defined by exactly one Cast\_part objects. Each Cast\_part defines the to\_be\_cast for exactly one Part\_relationship.

NOTE This assertion is established through the Part\_select.

#### **4.3.339 Part\_relationship to Cast\_part\_with\_rigging**

Each Part\_relationship has the to\_be\_cast defined by exactly one Cast\_part\_with\_rigging objects. Each Cast\_part\_with\_rigging defines the to\_be\_cast for exactly one Part\_relationship.

NOTE This assertion is established through the Part\_select.

#### **4.3.340 Part\_relationship to Design\_part**

Each Part\_relationship has the as\_designed defined by exactly one Design\_part objects. Each Design\_part defines the as\_designed for exactly one Part\_relationship.

#### **4.3.341 Part\_relationship to Tooling**

Each Part\_relationship has the tooling\_equipment defined by zero, one, or many Tooling objects. Each Tooling defines the tooling\_equipment for exactly one Part\_relationship.

#### **4.3.342 Part\_shape to Base\_shape**

Each Part\_shape has the base\_shape\_definition defined by zero or one Base\_shape objects. Each Base\_shape defines the base\_shape\_definition for zero, one, or many Part\_shape objects.

#### **4.3.343 Part\_shape to Object\_element\_shape\_representation**

Each Part\_shape has the representation\_form defined by zero, one, or many Object\_element\_shape\_representation objects. Each Object\_element\_shape\_representation defines the representation\_form for exactly one Part\_shape.

#### **4.3.344 Part\_shape to Shape\_aspect**

Each Part\_shape has the elements defined by zero, one, or many Shape\_aspect objects. Each Shape\_aspect defines the elements for zero, one, or many Part\_shape objects.

#### **4.3.345 Part\_version to Approval**

Each Part\_version \_version has the manufacture\_authorization defined by zero, one, or many Approval objects. Each Approval defines the manufacture\_authorization for zero, one, or many Part\_version objects.

#### **4.3.346 Part\_version to Customer\_casting\_requirement**

Each Part\_version has the customer\_requirement defined by zero, one, or many Customer\_casting\_requirement objects. Each Customer\_casting\_requirement defines the customer\_requirements for zero, one, or many Part\_version objects.

#### **4.3.347 Part\_version to Design\_reference**

Each Part\_version has the drawing\_info defined for zero or one Design\_reference objects. Each Design\_reference defines the drawing\_info for zero, one, or many Part\_version objects.

#### **4.3.348 Part\_version to External\_schema\_definition**

Each Part\_version has the casting\_defined\_by defined by zero or one External\_schema\_definition objects. Each External\_schema\_definition defines the casting\_defined\_by for zero, one or many Part\_version objects.

NOTE This assertion is established through process\_plan\_select.

#### **4.3.349 Part\_version to Material**

Each Part\_version has the material\_definition defined by one or more Material objects. Each Material defines the material\_definition for one or more Part\_version objects.

#### **4.3.350 Part\_version to Ordered\_part**

Each Part\_version has the quantity\_ordered defined by one or more Ordered\_part objects. Each Ordered\_part defines quantity\_ordered for exactly one Part\_version.

#### **4.3.351 Part\_version to Organization**

Each Part\_version has the manufactured\_by\_organization defined by one or more Organization objects. Each Organization defines the manufactured\_by\_organization for zero, one or many Part\_version objects.

#### **4.3.352 Part\_version to Part\_shape**

Each Part\_version has the physical\_form defined by exactly one Part\_shape. Each Part\_shape defines the physical\_form for zero, one, or many Part\_version objects.

Each Part\_version has the owned\_by\_organization defined by one or more Organization objects. Each Organization defines the owned\_by\_organization for zero, one, or many Part\_version objects.

NOTE This assertion is established through the person\_or\_organization\_select.

#### **4.3.353 Part\_version to Person\_in\_organization**

Each Part\_version has the manufactured\_by\_person defined by one or more Person\_in\_organization objects. Each Person\_in\_organization defines the manufactured\_by\_person for zero, one, or many Part\_version objects.

Each Part\_version has the owned\_by\_person defined by one or more Person\_in\_organization objects. Each Person\_in\_organization defines the owned\_by\_person for zero, one, or many Part\_version objects.

NOTE This assertion is established through the person\_or\_organization\_select.

#### **4.3.354 Part\_version to Process\_plan**

Each Part\_version has the casting\_defined\_by defined by zero or one Process\_plan objects. Each Process\_plan defines the casting\_defined\_by for zero, one or many Part\_version objects.

#### **4.3.355 Part\_version to Property**

Each Part\_version has the property\_characteristics defined by zero, one, or many Property objects. Each Property defines the applied\_to for exactly one Part\_version.

#### **4.3.356 Parting\_surface to Face\_shape\_element**

Each Parting\_surface has the shape\_element defined by one or more Face\_shape\_element. Each Face\_shape\_element defines the shape\_element for zero, one, or many Parting\_surface objects.

NOTE This assertion is established through the Shape\_element\_select.

#### **4.3.357 Parting\_surface to Numeric\_parameter**

Each Parting\_surface has the offset\_distance defined by one or more Numeric\_parameter. Each Numeric\_parameter defines the offset\_distance for zero, one, or many Parting\_surface objects.



#### 4.3.358 Parting\_surface to Planar\_element

Each Parting\_surface has the shape\_element defined by one or more Planar\_element. Each Planar\_element defines the shape\_element for zero, one, or many Parting\_surface objects.

NOTE This assertion is established through the Shape\_element\_select.

#### 4.3.359 Pattern\_equipment\_available to Flask

Each Pattern\_equipment\_available has the flask\_size\_for\_mounted\_patterns defined by zero or one Flask. Each Flask defines the flask\_size\_for\_mounted\_patterns for zero, one, or many Pattern\_equipment\_available objects.

Each Pattern\_equipment\_available has the flask\_size\_for\_plate\_patterns defined by zero or one Flask. Each Flask defines the flask\_size\_for\_plate\_patterns for zero, one, or many Pattern\_equipment\_available objects.

#### 4.3.360 Pattern\_equipment\_available to Flaskless

Each Pattern\_equipment\_available has the flaskless\_pattern defined by exactly one Flaskless. Each Flaskless defines the flaskless\_pattern for zero, one, or many Pattern\_equipment\_available objects.

#### 4.3.361 Pattern\_equipment\_available to In\_facility\_location

Each Pattern\_equipment\_available has the tool\_location defined by two or more In\_facility\_location. Each In\_facility\_location the tool\_location for zero, one, or many Pattern\_equipment\_available objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### 4.3.362 Pattern\_equipment\_available to Person\_in\_organization

Each Pattern\_equipment\_available has the tool\_location defined by two or more Person\_in\_organization. Each Person\_in\_organization the tool\_location for zero, one, or many Pattern\_equipment\_available objects.

NOTE This assertion is established through the Tool\_location\_select\_select.

#### 4.3.363 Pattern\_plate to Flask

Each Pattern\_plate has the defines\_flask\_type defined by exactly one Flask. Each Flask defines the defines\_flask\_type for zero, one, or many Pattern\_plate objects.

#### **4.3.364 Pattern\_plate to Gating\_system**

Each Pattern\_plate has the gate\_rigging defined by one or more Gating\_system. Each Gating\_system defines the gate\_rigging for zero, one, or many Pattern\_plate objects.

#### **4.3.365 Pattern\_plate to Item\_size**

Each Pattern\_plate has the pattern\_plate\_dimensions defined by exactly one Item\_size. Each Item\_size defines the pattern\_plate\_dimensions for zero, one, or many Pattern\_plate objects.

#### **4.3.366 Pattern\_plate to Location\_element**

Each Pattern\_plate has the mounting\_hole\_locations defined by one or more Location\_element. Each Location\_element defines the mounting\_hole\_locations for zero, one, or many Pattern\_plate objects.

#### **4.3.367 Pattern\_plate to Mould\_locks**

Each Pattern\_plate has the lock\_definitions defined by zero, one, or many Mould\_locks. Each Mould\_locks defines the lock\_definitions for zero, one, or many Pattern\_plate objects.

#### **4.3.368 Pattern\_plate to Moulding\_equipment**

Each Pattern\_plate has the moulding\_machine defined by exactly one Moulding\_equipment. Each Moulding\_equipment defines the moulding\_machine for zero, one, or many Pattern\_plate objects.

#### **4.3.369 Pattern\_plate to Parting\_surface**

Each Pattern\_plate has the parting\_line defined by exactly one Parting\_surface. Each Parting\_surface defines the parting\_line for zero, one, or many Pattern\_plate objects.

#### **4.3.370 Pattern\_plate to Pin\_center**

Each Pattern\_plate has the pin\_center\_definition defined by zero or one Pin\_center. Each Pin\_center defines the pin\_center\_definition for zero, one, or many Pattern\_plate objects.

#### **4.3.371 Permanent\_moulding\_process to Die\_mould\_vent**

Each Permanent\_moulding\_process has the vent\_definition defined by one or more Die\_mould\_vent. Each Die\_mould\_vent defines the vent\_definition for zero, one, or many Permanent\_moulding\_process objects.

#### **4.3.372 Permanent\_moulding\_process to Ejector\_system**

Each Permanent\_moulding\_process has the ejection\_definition defined by exactly one Ejector\_system. Each Ejector\_system defines the ejection\_definition for zero, one, or many Permanent\_moulding\_process objects.

#### **4.3.373 Permanent\_moulding\_process to Ingate**

Each Permanent\_moulding\_process has the metal\_flow\_system defined by one or more Ingate. Each Ingate defines the metal\_flow\_system for zero, one, or many Permanent\_moulding\_process objects.

NOTE This assertion is established through flow\_select.

#### **4.3.374 Permanent\_moulding\_process to Production\_die\_mould**

Each Permanent\_moulding\_process has the permanent\_die\_definition defined by exactly one Production\_die\_mould. Each Production\_die\_mould defines the permanent\_die\_definition for zero, one, or many Permanent\_moulding\_process objects.

#### **4.3.375 Permanent\_moulding\_process to Runner**

Each Permanent\_moulding\_process has the metal\_flow\_system defined by one or more Runner. Each Runner defines the metal\_flow\_system for zero, one, or many Permanent\_moulding\_process objects.

NOTE This assertion is established through flow\_select.

#### **4.3.376 Permanent\_moulding\_process to Sprue**

Each Permanent\_moulding\_process has the metal\_flow\_system defined by one or more Sprue. Each Sprue defines the metal\_flow\_system for zero, one, or many Permanent\_moulding\_process objects.

NOTE This assertion is established through flow\_select.

#### **4.3.377 Person to Address**

Each Person has the address defined by exactly one Address. Each Address defines the address for zero, one, or many Person objects.

#### **4.3.378 Person\_in\_organization to Organization**

Each Person\_in\_organization has the company defined by exactly one Organization. Each Organization defines the company for zero, one, or many Person\_in\_organization objects.

#### **4.3.379 Person\_in\_organization to Person**

Each Person\_in\_organization has the employee defined by exactly one Person. Each Person defines the employee for zero, one, or many Person\_in\_organization objects.

#### **4.3.380 Pin\_center to Numeric\_parameter\_with\_tolerance**

Each Pin\_center has the dimension defined by exactly one Numeric\_parameter\_with\_tolerance. Each Numeric\_parameter\_with\_tolerance defines the dimension for zero, one, or many Pin\_center objects.

#### **4.3.381 Pin\_tips to Shape\_element**

Each Pin\_tips has the shape defined by exactly one Shape\_element. Each Shape\_element defines the shape for zero, one, or many Pin\_tips objects.

#### **4.3.382 Placed\_target to Orientation**

Each Placed\_target has the placement defined by exactly one Orientation. Each Orientation defines placement for zero, one, or many Placed\_target objects.

#### **4.3.383 Planar\_element to Direction\_element**

Each Planar\_element has the normal defined by exactly one Direction\_element. Each Direction\_element defines the normal for zero, one, or many Planar\_element objects.

#### **4.3.384 Planar\_element to Location\_element**

Each Planar\_element has the location defined by exactly one Location\_element. Each Location\_element defines the location for zero, one, or many Planar\_element objects.

#### **4.3.385 Planar\_rib to Numeric\_parameter**

Each Planar\_rib has a part\_rib\_radius defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the part\_rib\_radius for zero, one, or many Planar\_rib objects.

Each Planar\_rib has a rib\_extent\_fillet defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the rib\_extent\_fillet for zero, one, or many Planar\_rib objects.

Each Planar\_rib has a transition\_to\_top\_radius defined by zero or one Numeric\_parameter. Each Numeric\_parameter defines the transition\_to\_top\_radius for zero, one, or many Planar\_rib objects.

Each `Planar_rib` has a `wall_to_top_radius` defined by exactly one `Numeric_parameter`. Each `Numeric_parameter` defines the `wall_to_top_radius` for zero, one, or many `Planar_rib` objects.

Each `Planar_rib` has a `wall_to_floor_fillet` defined by exactly one `Numeric_parameter`. Each `Numeric_parameter` defines the `wall_to_floor_fillet` to zero, one, or many `Planar_rib` objects.

#### **4.3.386 Planar\_rib to Face\_shape\_element**

Each `Planar_rib` has the top defined by one or more `Face_shape_element`. Each `Face_shape_element` defines the top for zero, one, or many `Planar_rib` objects.

NOTE This assertion is established through `Shape_select`.

#### **4.3.387 Planar\_rib to Planar\_element**

Each `Planar_rib` has the walls defined by one or more `Planar_element`. Each `Planar_element` defines the walls for zero, one, or many `Planar_rib` objects.

Each `Planar_rib` has the top defined by one or more `Planar_element`. Each `Planar_element` defines the top for zero, one, or many `Planar_rib` objects.

NOTE This assertion is established through `Shape_select`.

#### **4.3.388 Process\_parameter\_record to Property\_parameter**

Each `Process_parameter_record` has the `actual_parameter` defined by one or more `Property_parameter` objects. Each `Property_parameter` defines the `actual_parameter` for zero, one or many `Process_parameter_record` objects.

Each `Process_parameter_record` has the `planned_parameter` defined by one or more `Property_parameter` objects. Each `Property_parameter` defines the `planned_parameter` for zero, one or many `Process_parameter_record` objects.

#### **4.3.389 Process\_parameter\_record to Casting\_activity**

Each `Process_parameter_record` has the `recorded_process` defined by exactly one `Casting_activity` objects. Each `Casting_activity` defines the `recorded_process` for zero, one or many `Process_parameter_record` objects.

#### **4.3.390 Process\_plan to Activity\_execution\_result**

Each `Process_plan` has the `activity_results` defined by exactly one `Activity_execution_result` objects. Each `Activity_execution_result` defines the `activity_results` for zero, one or many `Process_plan` objects.

#### **4.3.391 Process\_plan to Casting\_activity**

Each Process\_plan has the activities defined by zero, one or many Casting\_activity objects. Each Casting\_activity defines the activities for zero, one or many Process\_plan objects.

#### **4.3.392 Process\_plan to Part\_version**

Each Process\_plan has the defines\_manufacturing\_for defined by exactly one Part\_version objects. Each Part\_version defines the defines\_manufacturing\_for for zero, one or many Process\_plan objects.

#### **4.3.393 Process\_plan\_security to Activity**

Each Process\_plan\_security has the identified\_by\_activity defined by exactly one Activity objects. Each Activity defines the identified\_by\_activity for zero, one or many Process\_plan\_security objects.

#### **4.3.394 Process\_plan\_security to Process\_plan\_version**

Each Process\_plan\_security has the identified\_by\_process\_plan defined by exactly one Process\_plan\_version objects. Each Process\_plan\_version defines the identified\_by\_process\_plan for zero, one or many Process\_plan\_security objects.

#### **4.3.395 Process\_plan\_version to Manufacturing\_process**

Each Process\_plan\_version has the activities\_to\_produce\_part defined by one or more Manufacturing\_process objects. Each Manufacturing\_process defines the activities\_to\_produce\_part for zero, one or many Process\_plan\_version objects.

#### **4.3.396 Process\_plan\_version to Material\_usage**

Each Process\_plan\_version has the amount\_of\_raw\_material defined by exactly one Material\_usage objects. Each Material\_usage defines the amount\_of\_raw\_material for zero, one or many Process\_plan\_version objects.

#### **4.3.397 Process\_plan\_version to Ordered\_part**

Each Process\_plan\_version has the quantity\_of\_parts defined by zero, one or many Ordered\_part objects. Each Ordered\_part defines the quantity\_of\_parts for zero, one or many Process\_plan\_version objects.

#### **4.3.398 Process\_plan\_version to Property\_parameter**

Each Process\_plan\_version has the process\_plan\_properties defined by zero, one or many Property\_parameter objects. Each Property\_parameter defines the process\_plan\_properties for zero, one or many Process\_plan\_version objects.

#### **4.3.399 Process\_plan\_version to Special\_instruction**

Each Process\_plan\_version has the process\_plan\_information defined zero, one or many Special\_instruction. Each Special\_instruction defines the process\_plan\_information for zero, one, or many Process\_plan\_version objects.

#### **4.3.400 Process\_property to Property\_parameter**

Each Process\_property has the property\_characteristic defined by zero, one, or many Property\_parameters objects. Each Property\_parameter defines the property characteristic for exactly one Process\_property.

#### **4.3.401 Process\_requirement to Descriptive\_parameter**

Each Process\_requirement defines the process\_name by exactly one Descriptive\_parameter. Each Descriptive\_parameter defines the process\_name by for zero, one, or many Process\_requirement objects.

#### **4.3.402 Product\_quality\_report to Inspection\_or\_test\_result**

Each Product\_quality\_report has the what\_failed defined by exactly one Inspection\_or\_test\_result objects. Each Inspection\_or\_test\_result defines the what\_failed for zero, one, or many Product\_quality\_report objects.

#### **4.3.403 Production\_core\_box to Casting\_insert**

Each Production\_core\_box has the inserts defined by zero, one or many Casting\_insert. Each Casting\_insert defines the inserts for zero, one, or many Production\_core\_box objects.

#### **4.3.404 Production\_core\_box to Chaplet\_pad**

Each Production\_core\_box has the chaplet\_pad\_definition defined by one or more Chaplet\_pad. Each Chaplet\_pad defines the chaplet\_pad\_definition for zero, one, or many Production\_core\_box objects.

#### **4.3.405 Production\_core\_box to Core**

Each Production\_core\_box has the core\_definition defined by one or more Core. Each Core defines the core\_definition for zero, one, or many Production\_core\_box objects.

#### **4.3.406 Production\_core\_box to Core\_equipment**

Each Production\_core\_box has the core\_made\_by defined by exactly one Core\_equipment. Each Core\_equipment defines the core\_made\_by for zero, one, or many Production\_core\_box objects.

#### **4.3.407 Production\_core\_box to Core\_print**

Each Production\_core\_box has the core\_print\_definition defined by zero, one or many Core\_print. Each Core\_print defines the core\_print\_definition for zero, one, or many Production\_core\_box objects.

#### **4.3.408 Production\_core\_box to Drafted\_surface**

Each Production\_core\_box has the draft defined by exactly one Drafted\_surface. Each Drafted\_surface defines the draft for zero, one, or many Production\_core\_box objects.

#### **4.3.409 Production\_core\_box to In\_facility\_location**

Each Production\_core\_box has the core\_box\_location defined by exactly one In\_facility\_location. Each In\_facility\_location defines the core\_box\_location for zero, one, or many Production\_core\_box objects.

NOTE This assertion is established through Tool\_location\_select.

#### **4.3.410 Production\_core\_box to Person\_in\_organization**

Each Production\_core\_box has the core\_box\_stored defined by exactly one Person\_in\_organization. Each Person\_in\_organization defines the core\_box\_stored for zero, one, or many Production\_core\_box objects.

NOTE This assertion is established through Tool\_location\_select.

#### **4.3.411 Production\_core\_box to Item\_size**

Each Production\_core\_box has the outside\_shape\_of\_core\_box defined by exactly one Item\_size. Each Item\_size defines the outside\_shape\_of\_core\_box for zero, one, or many Production\_core\_box objects.

Each Production\_core\_box has the guide\_pin defined by zero or one Item\_size. Each Item\_size defines the guide\_pin for zero, one, or many Production\_core\_box objects.

#### **4.3.412 Production\_core\_box to Loose\_piece**

Each Production\_core\_box has the loose\_piece\_requirement defined by zero, one or many Loose\_piece. Each Loose\_piece defines the loose\_piece\_requirement for zero, one, or many Production\_core\_box objects.

#### **4.3.413 Production\_core\_box to Machining\_allowance**

Each Production\_core\_box has the machine\_stock\_on\_tool defined by one or more Machining\_allowance. Each Machining\_allowance defines the machine\_stock\_on\_tool for zero, one, or many Production\_core\_box objects.



#### **4.3.414 Production\_core\_box to Material**

Each Production\_core\_box has the core\_box\_material defined by exactly one Material. Each Material defines the core\_box\_material for zero, one, or many Production\_core\_box objects.

#### **4.3.415 Production\_core\_box to Parting\_surface**

Each Production\_core\_box has the cope\_and\_drag\_parting defined by exactly one Parting\_surface. Each Parting\_surface defines the cope\_and\_drag\_parting for zero, one, or many Production\_core\_box objects.

Each Production\_core\_box has the offset\_parting\_surface defined by zero, one, or many Parting\_surface. Each Parting\_surface defines the offset\_parting\_surface for zero, one, or many Production\_core\_box objects.

#### **4.3.416 Production\_core\_box to Planar\_element**

Each Production\_core\_box has the vent\_placement defined by exactly one Planar\_element. Each Planar\_element defines the vent\_placement for zero, one, or many Production\_core\_box objects.

#### **4.3.417 Production\_core\_box to Shape\_aspect**

Each Production\_core\_box has the outside\_shape\_of\_core\_box defined by exactly one Shape\_aspect. Each Shape\_aspect defines the outside\_shape\_of\_core\_box for zero, one, or many Production\_core\_box objects.

NOTE This assertion is established through implicit\_or\_explicit\_shape.

#### **4.3.418 Production\_core\_box to Shape\_element**

Each Production\_core\_box has the impression\_shape defined by one or many Shape\_element. Each Shape\_element defines the impression\_shape for zero, one, or many Production\_core\_box objects.

Each Production\_core\_box has the invest\_area defined by zero, one, or many Shape\_element. Each Shape\_element defines the invest\_area for zero, one, or many Production\_core\_box objects.

#### **4.3.419 Production\_die\_cast\_mould to Cooling\_port**

Each Production\_die\_cast\_mould has the cooling\_type defined by zero or one Cooling\_port. Each Cooling\_port defines the cooling\_type for zero, one, or many Production\_die\_cast\_mould objects.

#### **4.3.420 Production\_die\_cast\_mould to Item\_size**

Each Production\_die\_cast\_mould has the external\_box\_dimensions defined by exactly one Item\_size. Each Item\_size defines the external\_box\_dimensions for zero, one, or many Production\_die\_cast\_mould objects.

#### **4.3.421 Production\_die\_cast\_mould to Die\_clamping**

Each Production\_die\_cast\_mould has the movable\_die\_items by zero, one, or many Die\_clamping. Each Die\_clamping defines the movable\_die\_items for zero, one, or many Production\_die\_cast\_mould objects.

NOTE This assertion is established through movable\_slide\_select.

#### **4.3.422 Production\_die\_cast\_mould to Shot\_sleeve**

Each Production\_die\_cast\_mould has the movable\_die\_items by zero, one, or many Shot\_sleeve. Each Shot\_sleeve defines the movable\_die\_items for zero, one, or many Production\_die\_cast\_mould objects.

NOTE This assertion is established through stationary\_die\_select.

#### **4.3.423 Production\_die\_cast\_mould to Runner**

Each Production\_die\_cast\_mould has the gate\_runner defined by one or more Runner. Each Runner defines the gate\_runner for zero, one, or many Production\_die\_cast\_mould objects.

#### **4.3.424 Production\_die\_cast\_mould to Secondary\_tooling**

Each Production\_die\_cast\_mould has the secondary\_tooling\_requirements defined by zero, one, or many Secondary\_tooling. Each Secondary\_tooling defines the secondary\_tooling\_requirements for zero, one, or many Production\_die\_cast\_mould objects.

#### **4.3.425 Production\_die\_cast\_mould to Shape\_element**

Each Production\_die\_cast\_mould has the internal\_shape defined by exactly one Shape\_element. Each Shape\_element defines the internal\_shape for zero, one, or many Production\_die\_cast\_mould objects.

#### **4.3.426 Production\_die\_cast\_mould to Slide**

Each Production\_die\_cast\_mould has the movable\_die\_items by zero, one, or many Slide. Each Slide defines the movable\_die\_items for zero, one, or many Production\_die\_cast\_mould objects.

NOTE This assertion is established through movable\_slide\_select.

#### **4.3.427 Production\_die\_cast\_mould to Core**

Each Production\_die\_cast\_mould has the core\_for\_die defined by exactly one Core. Each Core defines the core\_for\_die for zero, one, or many Production\_die\_cast\_mould objects.

**4.3.428 Production\_die\_cast\_mould to Die\_core**

Each Production\_die\_cast\_mould has the core\_for\_die defined by exactly one Die\_core. Each Die\_core defines the core\_for\_die for zero, one, or many Production\_die\_cast\_mould objects.

**4.3.429 Production\_die\_cast\_mould to Gating\_system**

Each Production\_die\_cast\_mould has the gating defined by exactly one Gating\_system. Each Gating\_system defines the gating for zero, one, or many Production\_die\_cast\_mould objects.

**4.3.430 Production\_die\_cast\_mould to Sprue**

Each Production\_die\_cast\_mould has the pouring\_sprue defined by exactly one Sprue. Each Sprue defines the pouring\_sprue for zero, one, or many Production\_die\_cast\_mould objects.

**4.3.431 Production\_die\_mould to Item\_size**

Each Production\_die\_mould has the external\_box\_dimensions defined by exactly one Item\_size. Each Item\_size defines the external\_box\_dimensions for zero, one, or many Production\_die\_mould objects.

**4.3.432 Production\_die\_mould to Slide**

Each Production\_die\_mould has the slide\_requirement defined by exactly one Slide. Each Slide defines the slide\_requirement for zero, one, or many Production\_die\_mould objects.

**4.3.433 Production\_investment\_cast\_mould to In\_facility\_location**

Each Production\_investment\_cast\_mould has the die\_mould\_storage defined by exactly one In\_facility\_location. Each In\_facility\_location defines the die\_mould\_storage for zero, one, or many Production\_investment\_cast\_mould objects.

NOTE This assertion is established through tool\_location\_select.

**4.3.434 Production\_investment\_cast\_mould to Person\_in\_organization**

Each Production\_investment\_cast\_mould has the die\_mould\_storage defined by exactly one Person\_in\_organization. Each Person\_in\_organization defines the die\_mould\_storage for zero, one, or many Production\_investment\_cast\_mould objects.

NOTE This assertion is established through tool\_location\_select.

**4.3.435 Production\_investment\_cast\_mould to Slide**

Each Production\_investment\_cast\_mould has the slide\_requirement by zero or one Slide. Each Slide defines the slide\_requirement for zero, one, or many Production\_investment\_cast\_mould objects.

#### **4.3.436 Production\_tool to Die\_core**

Each Production\_tool has the core\_for\_die defined by zero, one, or many Die\_core. Each Die\_core defines the core\_for\_die for zero, one, or many Production\_tool objects.

#### **4.3.437 Production\_tool to Drafted\_surface**

Each Production\_tool has the draft defined by one or more Drafted\_surface. Each Drafted\_surface defines the draft for zero, one, or many Production\_tool objects.

#### **4.3.438 Production\_tool to Ejector\_system**

Each Production\_tool has the ejection\_definition defined by exactly one Ejector\_system. Each Ejector\_system defines the ejection\_definition for zero, one, or many Production\_tool objects.

#### **4.3.439 Production\_tool to Item\_size**

Each Production\_tool has the guide\_pin defined by zero or one Item\_size. Each Item\_size defines the guide\_pin for zero, one, or many Production\_tool objects.

Each Production\_tool has the outside\_shape\_of\_mould defined by exactly one Item\_size. Each Item\_size defines the outside\_shape\_of\_mould for zero, one, or many Production\_tool objects.

NOTE This assertion is established through implicit\_or\_explicit\_select.

#### **4.3.440 Production\_tool to Machining\_allowance**

Each Production\_tool has the allowance\_on\_machinable\_surfaces defined by exactly one Machining\_allowance. Each Machining\_allowance defines the allowance\_on\_machinable\_surfaces for zero, one, or many Production\_tool objects.

#### **4.3.441 Production\_tool to Parting\_surface**

Each Production\_tool has the male\_or\_female defined by one or more Parting\_surface. Each Parting\_surface defines the male\_or\_female for zero, one, or many Production\_tool objects.

Each Production\_tool has the offset\_parting\_surfaces defined by zero, one, or many Parting\_surface. Each Parting\_surface defines the offset\_parting\_surfaces for zero, one, or many Production\_tool objects.

#### **4.3.442 Production\_tool to Production\_core\_box**

Each Production\_tool has the core\_definiton defined by zero, one, or many Production\_core\_box. Each Production\_core\_box defines the core\_definiton for zero, one, or many Production\_tool objects.

#### 4.3.443 Production\_tool to Riser

Each Production\_tool has the riser\_on\_mould defined by zero, one, or many Riser. Each Riser defines the riser\_on\_mould for zero, one, or many Production\_tool objects.

NOTE This assertion is established through riser\_select.

#### 4.3.444 Production\_tool to Riser\_contact

Each Production\_tool has the riser\_on\_mould defined by zero, one, or many Riser\_contact. Each Riser\_contact defines the riser\_on\_mould for zero, one, or many Production\_tool objects.

NOTE This assertion is established through riser\_select.

#### 4.3.445 Production\_tool to Shape\_aspect

Each Production\_tool has the outside\_shape\_of\_mould defined by exactly one Shape\_aspect. Each Shape\_aspect defines the outside\_shape\_of\_mould for zero, one, or many Production\_tool objects.

NOTE This assertion is established through implicit\_or\_explicit\_select.

#### 4.3.446 Production\_tool to Shape\_element

Each Production\_tool has the mould\_impression\_shape defined by zero, one, or many Shape\_element. Each Shape\_element defines the mould\_impression\_shape for zero, one, or many Production\_tool objects.

#### 4.3.447 Production\_tool to Vent

Each Production\_tool has the vent\_requirement defined by zero, one, or many Vent. Each Vent defines the vent\_requirement for zero, one, or many Production\_tool objects.

#### 4.3.448 Production\_pattern\_definition to Casting\_insert

Each Production\_pattern\_definition has the inserts defined by zero, one, or many Casting\_insert. Each Casting\_insert defines the inserts for zero, one, or many Production\_pattern\_definition objects.

#### 4.3.449 Production\_pattern\_definition to Chaplet\_pad

Each Production\_pattern\_definition has the chaplet\_pad\_definiton defined by zero, one or many Chaplet\_pad objects. Each Chaplet\_pad defines the chaplet\_pad\_definiton for zero, one or many Production\_pattern\_definition objects.

#### **4.3.450 Production\_pattern\_definition to Chill**

Each Production\_pattern\_definition has the chill\_pattern defined by exactly one Chill objects. Each Chill defines the chill\_pattern for zero, one or many Production\_pattern\_definition objects.

#### **4.3.451 Production\_pattern\_definition to Production\_core\_box**

Each Production\_pattern\_definition has the cores\_for\_pattern defined by zero, one or many Production\_core\_box objects. Each Production\_core\_box defines the cores\_for\_pattern for zero, one or many Production\_pattern\_definition objects.

#### **4.3.452 Production\_pattern\_definition to Core\_print**

Each Production\_pattern\_definition has the core\_print\_definition defined by zero, one or many Core\_print objects. Each Core\_print defines the core\_print\_definition for zero, one or many Production\_pattern\_definition objects.

#### **4.3.453 Production\_pattern\_definition to Drafted\_surface**

Each Production\_pattern\_definition has the draft by one or more Drafted\_surface objects. Each Drafted\_surface defines the draft\_angle for zero, one or many Production\_pattern\_definition objects.

#### **4.3.454 Production\_pattern\_definition to In\_facility\_location**

Each Production\_pattern\_definition has the pattern\_location by zero or one In\_facility\_location objects. Each In\_facility\_location defines the pattern\_location for zero, one or many Production\_pattern\_definition objects.

#### **4.3.455 Production\_pattern\_definition to Loose\_piece**

Each Production\_pattern\_definition has the loose\_piece\_requirement defined by zero, one or many Loose\_piece objects. Each Loose\_piece defines the loose\_piece\_requirement for zero, one or many Production\_pattern\_definition objects.

#### **4.3.456 Production\_pattern\_definition to Parting\_surface**

Each Production\_pattern\_definition has the offset\_parting\_surface defined by one or more Parting\_surface objects. Each Parting\_surface defines the offset\_parting\_surface for zero, one or many Production\_pattern\_definition objects.

Each Production\_pattern\_definition has the cope\_and\_drag\_parting defined by exactly one Parting\_surface objects. Each Parting\_surface defines the cope\_and\_drag\_parting for zero, one or many Production\_pattern\_definition objects.

#### **4.3.457 Production\_pattern\_definition to Pattern\_plate**

Each Production\_pattern\_definition has the pattern\_board\_size defined by zero or one Pattern\_plate objects. Each Pattern\_plate defines the pattern\_board\_size for zero, one or many Production\_pattern\_definition objects.

#### **4.3.458 Production\_pattern\_definition to Shape\_element**

Each Production\_pattern\_definition has the pattern\_shape by one or more Shape\_element objects. Each Shape\_element defines the pattern\_shape for zero, one or many Production\_pattern\_definition objects.

#### **4.3.459 Production\_pattern\_definition to Riser**

Each Production\_pattern\_definition has the riser\_on\_pattern defined by zero, one, or many Riser. Each Riser defines the riser\_on\_pattern for zero, one, or many Production\_pattern\_definition objects.

NOTE This assertion is established through riser\_select.

#### **4.3.460 Production\_pattern\_definition to Riser\_contact**

Each Production\_pattern\_definition has the riser\_on\_pattern defined by zero, one, or many Riser\_contact. Each Riser\_contact defines the riser\_on\_pattern for zero, one, or many Production\_pattern\_definition objects.

NOTE This assertion is established through riser\_select.

#### **4.3.461 Project\_order to Approval**

Each Project\_Order has the release\_authorization defined by exactly one Approval. Each Approval defines the release\_authorization for zero or one Project\_Order.

#### **4.3.462 Project\_order to Digital\_technical\_data\_package\_work\_order**

Each Project\_order has the technical\_data\_package\_status defined by zero or one Digital\_technical\_data\_package\_work\_order objects. Each Digital\_technical\_data\_package\_work\_order defines the technical\_data\_package\_status for exactly one Project\_order.

#### **4.3.463 Project\_order to Part\_version**

Each Project\_order has the part\_status defined by one or more Part\_version objects. Each Part\_version defines the part\_status for exactly one Project\_order.

#### **4.3.464 Project\_order to Pedigree\_creation\_order**

Each Project\_order has the pedigree\_creation\_status of zero or one Pedigree\_creation\_order. Each Pedigree\_creation\_order defines the pedigree\_creation\_status for exactly one Project\_order.

#### **4.3.465 Project\_order to Requisition**

Each Project\_order has the ordered\_resource defined by zero, one, or many Requisition objects. Each Requisition defines the ordered\_resource for exactly one Project\_order.

#### **4.3.466 Project\_order to Resource\_acquisition\_order**

Each Project\_order has the resource\_acquisition\_status defined by zero or one Resource\_acquisition\_order. Each Resource\_acquisition\_order defines the resource\_acquisition\_order for exactly one Project\_order.

#### **4.3.467 Project\_order to Shop\_work\_order**

Each Project\_order has the shop\_work\_status defined by of zero or one Shop\_work\_order. Each Shop\_work\_order defines the shop\_work\_status for exactly one Project\_order.

#### **4.3.468 Projection to Shape\_element**

Each Projection has the projection\_end defined by exactly one Shape\_element. Each Shape\_element defines the projection\_end for zero, one, or many Projection objects.

#### **4.3.469 Property to Casting\_weight\_requirement**

Each Property has the weight\_characteristics defined by exactly one Casting\_weight\_requirement objects. Each Casting\_weight\_requirement defines the weight\_characteristics for zero, one, or many Property objects.

#### **4.3.470 Property to Material\_property**

Each Property has the material\_characteristic defined by zero, one, or many Material\_property objects. Each Material\_property defines the material\_characteristic for zero, one, or many Property objects.

#### **4.3.471 Property to Part\_property**

Each Property has the part\_characteristic defined by zero, one, or many Part\_property objects. Each Part\_property defines the part\_characteristic for zero, one, or many Property objects.



**4.3.472 Property to Process\_property**

Each Property has the process\_characteristic defined by zero, one, or many Process\_property objects. Each Process\_property defines the process\_characteristic for zero, one, or many Property objects.

**4.3.473 Property to Shape\_aspect**

Each Property has the shape\_property\_characteristic defined by zero, one, or many Shape\_aspect objects. Each Shape\_aspect defines the shape\_property\_characteristic for zero, one, or many Property objects.

**4.3.474 Property to Specification**

Each Property has the property\_description defined by zero, one, or many Specification objects. Each Specification defines the property\_description for zero, one, or many Property objects.

**4.3.475 Property to Surface\_property**

Each Property has the surface\_characteristic defined by zero, one, or many Surface\_property objects. Each Surface\_property defines the process\_characteristic for zero, one, or many Property objects.

**4.3.476 Property\_BSU to Class\_BSU**

Each Property\_BSU has name\_scope defined by exactly one Class\_BSU. Each Class\_BSU defines name\_scope for zero, one, or many Property\_BSU objects.

**4.3.477 Property\_inspection to Property\_parameter**

Each Property\_inspection has the property\_result defined by zero, one, or many Property\_parameter objects. Each Property\_parameter defines the property\_result for zero, one, or many Property\_inspection objects.

Each Property\_inspection has the requirement defined by zero, one, or many Property\_parameter objects. Each Property\_parameter defines the requirement for zero, one, or many Property\_inspection objects.

**4.3.478 Property\_relationship to Data\_curve**

Each Property\_relationship has the relationship\_curve defined by zero, one, or many Data\_curve objects. Each Data\_curve defines the relationship\_curve for zero, one, or many Property\_relationship objects.

**4.3.479 Property\_value to Property\_BSU**

Each Property\_value has the property\_BSU defined by exactly one Property\_BSU objects. Each Property\_BSU defines the property\_BSU for zero, one, or many Property\_value objects.

#### **4.3.480 Quality\_acceptance\_report to Reporting\_requirement**

Each Quality\_acceptance\_report has the applied\_to defined by one or more Reporting\_requirement objects. Each Reporting\_requirement defines the applied\_to for zero, one, or many Quality\_acceptance\_report objects.

#### **4.3.481 Red\_line\_part to Feature**

Each Red\_line\_part has the mark\_up\_feature defined by zero, one, or many Feature objects. Each Feature defines the mark\_up\_feature for zero, one, or many Red\_line\_part objects.

#### **4.3.482 Red\_line\_part to Material**

Each Red\_line\_part has the mark\_up\_material defined by zero, one, or many Material objects. Each Material defines the mark\_up\_material for zero, one, or many Red\_line\_part objects.

#### **4.3.483 Red\_line\_part to Property**

Each Red\_line\_part has the mark\_up\_property defined by zero, one, or many Property objects. Each Property defines the mark\_up\_property for zero, one, or many Red\_line\_part objects.

#### **4.3.484 Red\_line\_part to Shape\_aspect**

Each Red\_line\_part has the mark\_up\_shape defined by zero, one, or many Shape\_aspect objects. Each Shape\_aspect defines the mark\_up\_shape for zero, one, or many Red\_line\_part objects.

#### **4.3.485 Reporting\_requirement to Date**

Each Reporting\_requirement has the date\_to\_be\_reported defined by exactly one Date objects. Each Date is the date\_to\_be\_reported for zero, one, or many Reporting\_requirement objects.

#### **4.3.486 Reporting\_requirement to Customer\_casting\_requirement**

Each Reporting\_requirement has the reporting\_based\_on defined by exactly one Customer\_casting\_requirement. Each Customer\_casting\_requirement defines the reporting\_based\_on for zero, one, or many Reporting\_requirement objects.

#### **4.3.487 Request\_for\_clarification to Person\_in\_organization**

Each Request\_for\_clarification has the requested\_by defined by exactly one Person\_in\_organization objects. Each Person\_in\_organization defines the requested\_by for zero, one, or many Request\_for\_clarification objects.

#### **4.3.488 Request\_for\_clarification to Part\_version**

Each Request\_for\_clarification has the requested\_part defined by exactly one Part\_version objects. Each Part\_version defines the requested\_part for zero, one, or many Request\_for\_clarification objects.

#### **4.3.489 Request\_for\_clarification to Date**

Each Request\_for\_clarification has the request\_date defined by exactly one Date objects. Each Date defines the request\_date for zero, one, or many Request\_for\_clarification objects.

#### **4.3.490 Request\_for\_quotation to Date**

Each Request\_for\_quotation has the order\_completion defined by exactly one Date objects. Each Date defines the order\_completion for zero, one, or many Request\_for\_quotation objects.

Each Request\_for\_quotation has the reply\_due\_date defined by exactly one Date objects. Each Date defines the reply\_due\_date for zero, one, or many Request\_for\_quotation objects.

Each Request\_for\_quotation has the delivery\_date defined by exactly one Date objects. Each Date defines the delivery\_date for zero, one, or many Request\_for\_quotation objects.

Each Request\_for\_quotation has the request\_date defined by exactly one Date objects. Each Date defines the request\_date for zero, one, or many Request\_for\_quotation objects.

#### **4.3.491 Request\_for\_quotation to Person\_in\_organization**

Each Request\_for\_quotation has the person\_submitting\_quote defined by exactly one Person\_in\_organization objects. Each Person\_in\_organization defines the person\_submitting\_quote for zero, one, or many Request\_for\_quotation objects.

#### **4.3.492 Request\_for\_quotation to Red\_line\_part**

Each Request\_for\_quotation has the redline defined by exactly one Red\_line\_part objects. Each Red\_line\_part defines the redline for zero, one, or many Request\_for\_quotation objects.

#### **4.3.493 Requisition to Date**

Each Requisition has the requisition\_date defined by exactly one Date objects. Each Date defines the requisition\_date for zero, one, or many Requisition objects.

Each Requisition has the required\_delivery\_date defined by exactly one Date objects. Each Date defines the required\_delivery\_date for zero, one, or many Requisition objects.

#### **4.3.494 Reset\_or\_return\_pins to Item\_size**

Each Reset\_or\_return\_pins has the pin\_dimensions defined by exactly one Item\_size objects. Each Item\_size defines the pin\_dimensions for zero, one, or many Reset\_or\_return\_pins objects.

#### **4.3.495 Resource\_with\_material to Material**

Each Resource\_with\_material has the resource\_material defined by exactly one Material objects. Each Material defines the resource\_material for zero, one, or many Resource\_with\_material objects.

#### **4.3.496 Resource\_with\_representation to Externally\_defined\_representation**

Each Resource\_with\_representation has the resource\_documentation defined by exactly one Externally\_defined\_representation objects. Each Externally\_defined\_representation defines the resource\_documentation for zero, one, or many Resource\_with\_representation objects.

#### **4.3.497 Revision to Status\_authority**

Each Revision has the approved\_by defined by exactly one Status\_authority objects. Each Status\_authority defines the approved\_by for zero, one, or many Revision objects.

#### **4.3.498 Revision to Process\_plan\_version**

Each Revision has the related\_to defined by exactly one Process\_plan\_version objects. Each Process\_plan\_version defines the related\_to for zero, one, or many Revision objects.

Each Revision has the relating\_to defined by exactly one Process\_plan\_version objects. Each Process\_plan\_version defines the relating\_to for zero, one, or many Revision objects.

#### **4.3.499 Riser\_contact to Shape\_element**

Each Riser\_contact has the break\_off\_connection defined by exactly one Shape\_element objects. Each Shape\_element defines the break\_off\_connection for zero, one, or many Riser\_contact objects.

#### **4.3.500 Risers to Item\_size**

Each Risers has the riser\_dimensions defined by exactly one Item\_size objects. Each Item\_size defines the riser\_dimensions for zero, one, or many Risers objects.

Each Risers has the riser\_neck defined by exactly one Item\_size objects. Each Item\_size defines the riser\_neck for zero, one, or many Risers objects.

Each Risers has the riser\_core defined by zero or one Item\_size objects. Each Item\_size defines the riser\_core for zero, one, or many Risers objects.

#### **4.3.501 Risers to Riser\_contact**

Each Risers has the contact\_definition defined by zero, one, or many Riser\_contact objects. Each Riser\_contact defines the contact\_definition for zero, one, or many Risers objects.

#### **4.3.502 Risers to Well**

Each Risers has the riser\_well defined by zero, one, or many Well objects. Each Well defines the riser\_well for zero, one, or many Risers objects.

#### **4.3.503 Runners to Choke**

Each Runners has the runner\_choke defined by zero or one Choke objects. Each Choke defines the runner\_choke for zero, one, or many Runners objects.

#### **4.3.504 Runners to Feeder**

Each Runners has the riser\_connections defined by zero, one, or many Feeders objects. Each Feeders defines the riser\_connections for zero, one, or many Runners objects.

#### **4.3.505 Runners to Filter**

Each Runners has the metal\_flow\_filter defined by zero, one, or many Filter objects. Each Filter defines the metal\_flow\_filter for zero, one, or many Runners objects.

#### **4.3.506 Runners to Item\_size**

Each Runners has the runner\_dimensions defined by exactly one Item\_size objects. Each Item\_size defines the runner\_dimensions for zero, one, or many Runners objects.

#### **4.3.507 Runners to Sprue**

Each Runners has the sprue\_connections defined by one or more Sprue objects. Each Sprue defines the sprue\_connections for zero, one, or many Runners objects.

#### **4.3.508 Runners to Vent**

Each Runners has the runner\_vent defined by zero, one, or many Vent objects. Each Vent defines the runner\_vent for zero, one, or many Runners objects.

#### **4.3.509 Sampled\_set to Product\_quality\_report**

Each Sampled\_set has the quality\_assurance\_document defined by one or more Product\_quality\_report. Each Product\_quality\_report defines the quality\_assurance\_document for zero, one, or many Sampled\_set objects.

#### **4.3.510 Sampled\_set to Master\_sample**

Each Sampled\_set has the master\_samples\_inspected defined by exactly one Master\_sample objects. Each Master\_sample defines the master\_samples\_inspected for zero, one, or many Sampled\_set objects.

#### **4.3.511 Sampled\_set to Heat**

Each Sampled\_set has the sampled\_from defined by exactly one Heat objects. Each Heat defines the sampled\_from for zero, one, or many Sampled\_set objects.

NOTE This assertion is established through Sample\_form.

#### **4.3.512 Sampled\_set to Lot**

Each Sampled\_set has the sampled\_from defined by exactly one Lot objects. Each Lot defines the sampled\_from for zero, one, or many Sampled\_set objects.

NOTE This assertion is established through Sample\_form.

#### **4.3.513 Sand\_casting\_design\_feature to Orientation**

Each Sand\_casting\_design\_feature has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Sand\_casting\_design\_feature objects.

#### **4.3.514 Sand\_casting\_tooling to Production\_pattern\_definition**

Each Sand\_casting\_tooling has the pattern\_equipment defined by zero, one or many Production\_pattern\_definition objects. Each Production\_pattern\_definition defines the pattern\_equipment for zero, one or many Sand\_casting\_tooling objects.

#### **4.3.515 Sand\_mould to Equipment**

Each Sand\_mould has the mould\_made\_by defined by exactly one Equipment. Each Equipment defines the mould\_made\_by for zero, one, or many Sand\_mould objects.

#### **4.3.516 Sand\_mould to Chaplet**

Each Sand\_mould has the chaplet\_definition by exactly one Chaplet. Each Chaplet defines the chaplet\_definition for zero, one, or many Sand\_mould objects.

**4.3.517 Sand\_mould to Chill**

Each Sand\_mould has the chill defined by exactly one Chill. Each Chill defines the chill for zero, one, or many Sand\_mould objects.

**4.3.518 Shape\_aspect to Shape\_element**

Each Shape\_aspect has the element defined by zero or one Shape\_element. Each Shape\_element defines the element for exactly one Shape\_aspect.

**4.3.519 Shape\_aspect to Model\_element**

Each Shape\_aspect has the representation\_shape defined by zero, one, or many Model\_element objects. Each Model\_element defines the representation\_shape for exactly one Shape\_aspect.

**4.3.520 Shape\_aspect to Shape\_aspect\_representation**

Each Shape\_aspect has the representation\_form defined by one, or more Shape\_aspect\_representation objects. Each Shape\_aspect\_representation defines the representation\_form for exactly one Shape\_aspect.

**4.3.521 Shape\_aspect\_representation to Geometric\_model**

Each Shape\_aspect\_representation has the shape\_definition defined by exactly one Geometric\_model. Each Geometric\_model defines the shape\_definition for exactly one Shape\_aspect\_representation.

**4.3.522 Shape\_dimensions to Production\_core\_box**

Each Shape\_dimensions has the tooling defined by exactly one Production\_core\_box. Each Production\_core\_box defines the tooling for zero, one or many Shape\_dimensions objects.

NOTE This assertion is established through tooling\_select.

**4.3.523 Shape\_dimensions to Production\_die\_mould**

Each Shape\_dimensions has the tooling defined by exactly one Production\_die\_mould. Each Production\_die\_mould defines the tooling for zero, one or many Shape\_dimensions objects.

NOTE This assertion is established through tooling\_select.

#### **4.3.524 Shape\_dimensions to Production\_die\_cast\_mould**

Each Shape\_dimensions has the tooling defined by exactly one Production\_die\_cast\_mould. Each Production\_die\_cast\_mould defines the tooling for zero, one or many Shape\_dimensions objects.

NOTE This assertion is established through tooling\_select.

#### **4.3.525 Shape\_dimensions to Production\_investment\_cast\_mould**

Each Shape\_dimensions has the tooling defined by exactly one Production\_investment\_cast\_mould. Each Production\_investment\_cast\_mould defines the tooling for zero, one or many Shape\_dimensions objects.

NOTE This assertion is established through tooling\_select.

#### **4.3.526 Shape\_dimensions to Production\_pattern\_definition**

Each Shape\_dimensions has the tooling defined by exactly one Production\_pattern\_definition. Each Production\_pattern\_definition the tooling for zero, one or many Shape\_dimensions objects.

NOTE This assertion is established through tooling\_select.

#### **4.3.527 Shot\_sleeve to Item\_size**

Each Shot\_sleeve has the slot\_sleeve\_size defined by exactly one Item\_size objects. Each Item\_size defines the slot\_sleeve\_size for zero, one, or many Shot\_sleeve objects.

#### **4.3.528 Simulation\_input to Condition\_or\_assumption**

Each Simulation\_input has the assumption defined by zero, one, or many Condition\_or\_assumption. Each Condition\_or\_assumption defines the assumption for zero, one, or many Simulation\_input objects.

Each Simulation\_input has the condition defined by zero, one, or many Condition\_or\_assumption. Each Condition\_or\_assumption defines the condition for zero, one, or many Simulation\_input objects.

#### **4.3.529 Simulation\_input to Integration\_interval**

Each Simulation\_input has the integration\_time\_specification defined by one or more Integration\_interval. Each Integration\_interval defines the integration\_time\_specification for zero, one, or many Simulation\_input objects.

#### **4.3.530 Simulation\_input to Document\_assignment**

Each Simulation\_input has the setup\_file defined by exactly one Document\_assignment. Each Document\_assignment defines the setup\_file for zero, one, or many Simulation\_input objects.



#### **4.3.531 Simulation\_input to Numeric\_parameter**

Each Simulation\_input has the input\_parameter defined by one or more Numeric\_parameter objects. Each Numeric\_parameter defines the input\_parameter for zero, one or many Simulation\_input objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.532 Simulation\_input to Property\_relationship**

Each Simulation\_input has the input\_parameter defined by exactly one Property\_relationship objects. Each Property\_relationship defines the input\_parameter for zero, one or many Simulation\_input objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.533 Simulation\_input to Simulation\_input\_region**

Each Simulation\_input has the region defined by exactly one Simulation\_input\_region. Each Simulation\_input\_region defines the region for zero, one, or many Simulation\_input objects.

#### **4.3.534 Simulation\_input\_region to Boundary\_condition**

Each Simulation\_input\_region has the boundary\_condition defined by zero, one or many Boundary\_condition objects. Each Boundary\_condition defines the boundary\_condition for zero, one or many Simulation\_input\_region objects.

#### **4.3.535 Simulation\_input\_region to Meshing\_condition**

Each Simulation\_input\_region has the meshing\_condition defined by zero, one or many Meshing\_condition objects. Each Meshing\_condition defines the meshing\_condition for zero, one or many Simulation\_input\_region objects.

#### **4.3.536 Simulation\_input\_region to Numeric\_parameter**

Each Simulation\_input\_region has the region\_property defined by one or more Numeric\_parameter objects. Each Numeric\_parameter defines the region\_property for zero, one or many Simulation\_input\_region objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.537 Simulation\_input\_region to Part\_version**

Each Simulation\_input\_region has the associated\_part defined by one or more Part\_version objects. Each Part\_version defines the associated\_part for zero, one or many Simulation\_input\_region objects.

#### **4.3.538 Simulation\_input\_region to Property\_relationship**

Each Simulation\_input\_region has the region\_property defined by exactly one Property\_relationship objects. Each Property\_relationship defines the region\_property for zero, one or many Simulation\_input\_region objects.

NOTE This assertion is established through property\_value\_or\_relationship\_select.

#### **4.3.539 Simulation\_input\_region to Shape\_element**

Each Simulation\_input\_region has the simulation\_region\_shape defined by exactly one Shape\_element objects. Each Shape\_element defines the simulation\_region\_shape for zero, one or many Simulation\_input\_region objects.

#### **4.3.540 Simulation\_output to Defect\_prediction**

Each Simulation\_output has the predicted\_defect defined by zero, one or many Defect\_prediction objects. Each Defect\_prediction defines the predicted\_defect for zero, one or many Simulation\_output objects.

#### **4.3.541 Simulation\_output to Simulation\_result**

Each Simulation\_output has the result defined by zero, one or many Simulation\_result objects. Each Simulation\_result defines the result for zero, one or many Simulation\_output objects.

#### **4.3.542 Simulation\_output\_region to Shape\_element**

Each Simulation\_output\_region has the region\_shape defined by exactly one Shape\_element objects. Each Shape\_element defines the region\_shape for zero, one or many Simulation\_output\_region objects.

#### **4.3.543 Simulation\_output\_region to Simulation\_input\_region**

Each Simulation\_output\_region has the belongs\_to defined by exactly one Simulation\_input\_region objects. Each Simulation\_input\_region defines the belongs\_to for zero, one or many Simulation\_output\_region objects.

#### **4.3.544 Simulation\_output\_region to Simulation\_unit\_state**

Each Simulation\_output\_region has the history defined by one or more Simulation\_unit\_state objects. Each Simulation\_unit\_state defines the history for zero, one or many Simulation\_output\_region objects.

#### **4.3.545 Simulation\_process to Design\_part**

Each Simulation\_process has the part\_geomerty defined by exactly one Design\_part objects. Each Design\_part defines the part\_geomerty for zero, one or many Simulation\_process objects.

#### **4.3.546 Simulation\_process to Cast\_part**

Each Simulation\_process has the part\_geomerty defined by exactly one Cast\_part objects. Each Cast\_part defines the part\_geomerty for zero, one or many Simulation\_process objects.

#### **4.3.547 Simulation\_process to Cast\_part\_with\_rigging**

Each Simulation\_process has the part\_geomerty defined by exactly one Cast\_part\_with\_rigging objects. Each Cast\_part\_with\_rigging defines the part\_geomerty for zero, one or many Simulation\_process objects.

#### **4.3.548 Simulation\_process to Customer\_casting\_requirement**

Each Simulation\_process has the quality\_requirements defined by one or more Customer\_casting\_requirement objects. Each Customer\_casting\_requirement defines the quality\_requirements for zero, one or many Simulation\_process objects.

#### **4.3.549 Simulation\_process to Simulation\_run**

Each Simulation\_process has the run\_simulation defined by exactly one Simulation\_run objects. Each Simulation\_run defines the run\_simulation for zero, one or many Simulation\_process objects.

#### **4.3.550 Simulation\_process to Tooling**

Each Simulation\_process has the tooling\_for\_simulation defined by exactly one Tooling objects. Each Tooling defines the tooling\_for\_simulation for zero, one or many Simulation\_process objects.

#### **4.3.551 Simulation\_result to Illustration**

Each Simulation\_result has the illustration defined by zero, one or many Illustration objects. Each Illustration defines the illustration for zero, one or many Simulation\_result objects.

#### **4.3.552 Simulation\_result to Simulation\_output\_region**

Each Simulation\_result has the region defined by one or more Simulation\_output\_region objects. Each Simulation\_output\_region defines the region for zero, one or many Simulation\_result objects.

#### **4.3.553 Simulation\_run to Activity**

Each Simulation\_run has the associated\_process defined by exactly one Activity objects. Each Activity defines the associated\_process for zero, one or many Simulation\_run objects.

#### **4.3.554 Simulation\_run to Simulation\_input**

Each Simulation\_run has the input defined by exactly one Simulation\_input objects. Each Simulation\_input defines the input for zero, one or many Simulation\_run objects.

#### **4.3.555 Simulation\_run to Simulation\_software**

Each Simulation\_run has the software\_used defined by exactly one Simulation\_software objects. Each Simulation\_software defines the software\_used for zero, one or many Simulation\_run objects.

#### **4.3.556 Simulation\_run to Simulation\_output**

Each Simulation\_run has the output defined by exactly one Simulation\_output objects. Each Simulation\_output defines the output for zero, one or many Simulation\_run objects.

#### **4.3.557 Simulation\_run\_relationship to Simulation\_run**

Each Simulation\_run\_relationship has the related defined by exactly one Simulation\_run objects. Each Simulation\_run defines the related for zero, one or many Simulation\_run\_relationship objects.

Each Simulation\_run\_relationship has the relating defined by exactly one Simulation\_run objects. Each Simulation\_run defines the relating for zero, one or many Simulation\_run\_relationship objects.

#### **4.3.558 Simulation\_software to Date**

Each Simulation\_software has the release\_date defined by exactly one Date objects. Each Date defines the release\_date for zero, one or many Simulation\_run objects.

#### **4.3.559 Simulation\_unit to Location\_element**

Each Simulation\_unit has the unit\_location defined by exactly one Location\_element objects. Each Location\_element defines the unit\_location for zero, one or many Simulation\_unit objects.

#### **4.3.560 Simulation\_unit to Simulation\_output\_region**

Each Simulation\_unit has the belongs\_to defined by exactly one Simulation\_output\_region objects. Each Simulation\_output\_region defines the belongs\_to for zero, one or many Simulation\_unit objects.

#### **4.3.561 Simulation\_unit to Simulation\_unit\_state**

Each Simulation\_unit has the history defined by one or more Simulation\_unit\_state objects. Each Simulation\_unit\_state defines the history for zero, one or many Simulation\_unit objects.

**4.3.562 Simulation\_unit to Shape\_aspect**

Each Simulation\_unit has the unit\_shape defined by exactly one Shape\_aspect objects. Each Shape\_aspect defines the unit\_shape for zero, one or many Simulation\_unit objects.

**4.3.563 Simulation\_unit\_state to Simulated\_property**

Each Simulation\_unit\_state has the property defined by exactly one Simulated\_property objects. Each Simulated\_property defines the property for zero, one or many Simulation\_unit\_state objects.

**4.3.564 Size\_tolerance to Shape\_aspect**

Each Size\_tolerance has the applied\_shape defined exactly one Shape\_aspect. Each Shape\_aspect defines the applied\_shape for zero, one, or many Size\_dimension objects.

**4.3.565 Slide to Shape\_element**

Each Slide has the shape defined by one or more Shape\_element objects. Each Shape\_element defines the shape for zero, one or many Slide objects.

**4.3.566 Special\_inspection\_requirement to Specification**

Each Special\_inspection\_requirement has the inspection\_requirements defined by exactly one Specification objects. Each Specification defines the inspection\_requirements for zero, one, or many Special\_inspection\_requirement.

**4.3.567 Specification to Date**

Each Specification has the release\_date defined by exactly one Date objects. Each Date defines the release\_date for zero, one, or many Specification.

**4.3.568 Specification to Specification\_usage\_constraint**

Each Specification has the constraint defined by zero, one, or many Specification\_usage\_constraint objects. Each Specification\_usage\_constraint defines the constraint for exactly one Specification.

**4.3.569 Specification to Organization**

Each Specification has the issuing\_organization defined by exactly one Organization objects. Each Organization defines the issuing\_organization for zero, one, or many Specification.

#### **4.3.570 Specification to Document\_assignment**

Each Specification has the specification\_document defined by zero or one Document\_assignment. Each Document\_assignment defines the specification\_document for zero, one, or many Specification objects.

#### **4.3.571 Sprue to Filter**

Each Sprue has the metal\_flow\_filter defined by zero or one Filter. Each Filter defines the metal\_flow\_filter for zero, one, or many Sprue objects.

#### **4.3.572 Sprue to Item\_size**

Each Sprue has the sprue\_dimensions defined by exactly one Item\_size. Each Item\_size defines the sprue\_dimensions for zero, one, or many Sprue objects.

Each Sprue has the pouring\_cup defined by zero, one, or many Item\_size. Each Item\_size defines the pouring\_cups for zero, one, or many Sprue objects.

#### **4.3.573 Sprue to Well**

Each Sprue has the sprue\_dimensions defined by zero, one, or many Well. Each Well defines the sprue\_dimensions for zero, one, or many Sprue objects.

#### **4.3.574 Sprue\_and\_runner\_mould to Ingate**

Each Sprue\_and\_runner\_mould has the rigging\_component defined by one or more Ingate. Each Ingate defines the rigging\_component for zero, one, or many Sprue\_and\_runner\_mould objects.

#### **4.3.575 Sprue\_and\_runner\_mould to Investment\_mould**

Each Sprue\_and\_runner\_mould has the mould\_assembly defined by exactly one Investment\_mould. Each Investment\_mould defines the mould\_assembly for zero, one, or many Sprue\_and\_runner\_mould objects.

#### **4.3.576 Sprue\_and\_runner\_mould to Runner**

Each Sprue\_and\_runner\_mould has the gate\_runner defined by one or more Runner. Each Runner defines the gate\_runner for zero, one, or many Sprue\_and\_runner\_mould objects.

#### **4.3.577 Sprue\_and\_runner\_mould to Sprue**

Each Sprue\_and\_runner\_mould has the pouring\_sprue defined by exactly one Sprue. Each Sprue defines the pouring\_sprue for zero, one, or many Sprue\_and\_runner\_mould objects.

#### **4.3.578 Status\_authority to Person\_in\_organization**

Each Status\_authority has the is\_given\_by defined by exactly one Person\_in\_organization. Each Person\_in\_organization defines the is\_given\_by for zero, one, or many Status\_authority objects.

#### **4.3.579 Stop\_pin to Item\_size**

Each Stop\_pin has the size defined by exactly one Item\_size. Each Item\_size defines the size for zero, one, or many Stop\_pin objects.

#### **4.3.580 Surface\_property to Property\_parameter**

Each Surface\_property has the property\_characteristic defined by zero, one, or many Property\_parameter objects. A Property\_parameter defines the property\_characteristic for exactly one Surface\_property.

#### **4.3.581 Target\_area to Shape\_element**

Each Target\_area has the area\_shape defined by exactly one Shape\_element. Each Shape\_element defines the area\_shape by zero, one, or many Shape\_element objects.

#### **4.3.582 Thickness\_tolerance to Shape\_aspect**

Each Thickness\_tolerance has the applied\_shape defined zero or one Shape\_aspect. Each Shape\_aspect defines the applied\_shape for zero, one, or many Thickness\_tolerance objects.

#### **4.3.583 Tolerance\_inspection to Tolerance\_requirement**

Each Tolerance\_inspection has the requirement defined by one or more Tolerance\_requirement. Each Tolerance\_requirement defines the requirement for zero, one, or many Tolerance\_inspection objects.

#### **4.3.584 Tolerance\_inspection to Numeric\_parameter**

Each Tolerance\_inspection has the tolerance\_result defined by exactly one Numeric\_parameter. Each Numeric\_parameter defines the tolerance\_result for zero, one, or many Tolerance\_inspection objects.

NOTE This assertion is established through toleranace\_select.

#### **4.3.585 Tolerance\_requirement to Mismatch\_tolerance**

Each Tolerance\_requirement has the tolerance\_based\_on defined by exactly one Mismatch\_tolerance. Each Mismatch\_tolerance defines the tolerance\_based\_on for zero, one, or many Tolerance\_requirement objects.

#### **4.3.586 Tolerance\_requirement to Checking\_aid\_tool**

Each Tolerance\_requirement has the gaging\_requirement defined by zero or one Checking\_aid\_tool. Each Checking\_aid\_tool defines the gaging\_requirement for zero, one, or many Tolerance\_requirement objects.

#### **4.3.587 Tolerance\_requirement to Specification**

Each Tolerance\_requirement has the dimension\_specification defined by zero or one Specification. Each Specification defines the dimension\_specification for zero, one, or many Tolerance\_requirement objects.

#### **4.3.588 Tolerance\_value to Limits\_and\_fits**

Each Tolerance\_value has the defined\_value by zero or one Limits\_and\_fits. Each Limits\_and\_fits defines the defined\_value for zero, one, or many Tolerance\_value objects.

NOTE This assertion is established through Tolerance\_definition\_select.

#### **4.3.589 Tolerance\_value to Plus\_minus\_value**

Each Tolerance\_value has the defined\_value by zero or one Plus\_minus\_value. Each Plus\_minus\_value defines the defined\_values for zero, one, or many Tolerance\_value objects.

NOTE This assertion is established through Tolerance\_definition\_select.

#### **4.3.590 Tolerance\_value to Tolerance\_limit**

Each Tolerance\_value has the defined\_value by zero or one tolerance\_limit. Each tolerance\_limit defines the defined\_value for zero, one, or many Tolerance\_value objects.

NOTE This assertion is established through Tolerance\_definition\_select.

#### **4.3.591 Tolerance\_value to Tolerance\_range**

Each Tolerance\_value has the defined\_value by zero or one tolerance\_range. Each tolerance\_range defines the defined\_value for zero, one, or many Tolerance\_value objects.

NOTE This assertion is established through Tolerance\_definition\_select.

#### **4.3.592 Tolerance\_zone to Tolerance\_zone\_definition**

Each Tolerance\_zone has zone\_definition defined by exactly one Tolerance\_zone\_definition. Each Tolerance\_zone\_definition defines zone\_definition for zero, one, or many Tolerance\_zone objects.



**4.3.593 Tolerance\_zone to Projection**

Each Tolerance\_zone has extended\_shape defined by zero, one, two, or three Projection objects. Each Projection defines extended\_shape for zero, one, or many Tolerance\_zone objects.

**4.3.594 Tolerance\_zone\_definition to Shape\_element**

Each Tolerance\_zone\_definition has the first\_element defined by exactly one Shape\_element. Each Shape\_element defines the first\_element for zero, one, or many Tolerance\_zone\_definition objects.

Each Tolerance\_zone\_definition has the second\_element defined by zero or one Shape\_element. Each Shape\_element defines the second\_element for zero, one, or many Tolerance\_zone\_definition objects.

**4.3.595 Tool\_verification to Core\_assembly**

Each Tool\_verification has the core\_fit defined by zero or one Core\_assembly. Each Core\_assembly defines the core\_fit for zero, one, or many Tool\_verification objects.

**4.3.596 Tool\_verification to Core\_print**

Each Tool\_verification has the verify\_clearance defined by exactly one Core\_print. Each Core\_print defines the verify\_clearance for zero, one, or many Tool\_verification objects.

**4.3.597 Tool\_verification to Design\_part**

Each Tool\_verification has the based\_on\_part\_design defined by exactly one Design\_part. Each Design\_part defines the based\_on\_part\_design for zero, one, or many Tool\_verification objects.

**4.3.598 Tool\_verification to Shape\_dimensions**

Each Tool\_verification has the verify defined by exactly one Shape\_dimensions. Each Shape\_dimensions defines the verify for zero, one, or many Tool\_verification objects.

**4.3.599 Tool\_verification to Tolerance\_requirement**

Each Tool\_verification has the customer\_requirement defined by exactly one Tolerance\_requirement. Each Tolerance\_requirement defines the customer\_requirement for zero, one, or many Tool\_verification objects.

**4.3.600 Tooling to Cast\_part**

Each Tooling has the tooling\_designed\_for defined by exactly one Cast\_part objects. Each Cast\_part defines the tooling\_designed\_for for zero, one or many Tooling objects.

#### **4.3.601 Tooling to Date**

Each Tooling has the `date_last_used` defined by exactly one Date objects. Each Date defines the `date_last_used` for zero, one or many Tooling objects.

#### **4.3.602 Tooling\_process to Core**

Each Tooling\_process has the `cores_to_make` defined by zero, one or many Core objects. Each Core defines the `cores_to_make` for zero, one or many Tooling\_process objects.

#### **4.3.603 Tooling\_process to Customer\_casting\_requirement**

Each Tooling\_process has the `customer_requirements` defined by exactly one customer\_casting\_requirement objects. Each customer\_casting\_requirement defines the `customer_requirements` for zero, one or many Tooling\_process objects.

#### **4.3.604 Tooling\_process to Design\_part**

Each Tooling\_process has the `type_of_tooling` defined by exactly one Design\_part objects. Each Design\_part defines the `type_of_tooling` for zero, one or many Tooling\_process objects.

#### **4.3.605 Tooling\_process to Gating\_system**

Each Tooling\_process has the `required_gating` defined by exactly one Gating\_system objects. Each Gating\_system defines the `required_gating` for zero, one or many Tooling\_process objects.

#### **4.3.606 Tooling\_process to Tooling**

Each Tooling\_process has the `type_of_tooling` defined by exactly one Tooling objects. Each Tooling defines the `type_of_tooling` for zero, one or many Tooling\_process objects.

#### **4.3.607 Tooling\_process to Tooling\_verification**

Each Tooling\_process has the `quality_assurance` defined by exactly one Tooling\_verification objects. Each Tooling\_verification defines the `quality_assurance` for zero, one or many Tooling\_process objects.

#### **4.3.608 Vent to Item\_size**

Each Vent has the `Vent_size` defined by exactly one Item\_size. Each Item\_size defines the `Vent_size` for zero, one, or many Vent objects.

#### **4.3.609 Vent to Location\_element**

Each Vent has the `Vent_placement` defined by zero, one, or many Location\_element. Each Location\_element defines the `Vent_placement` for zero, one, or many Vent objects.

### 4.3.610 View\_reference to Design\_reference

Each View\_reference has the owned\_by defined by exactly one Design\_reference. Each Design\_reference defines the owned\_by for zero, one, or many View\_reference objects.

### 4.3.611 Well to Item\_size

Each Well has the well\_dimensions defined by exactly one Item\_size. Each Item\_size defines the well\_dimensions for zero, one, or many Well objects.

### 4.3.612 Well to Parting\_surface

Each Well has the drag\_surface defined exactly one Parting\_surface. Each Parting\_surface defines the drag\_surface for zero, one, or many Well objects.

### 4.3.613 Width\_dimension to Shape\_aspect

Each Width\_dimension has the applied\_shape defined zero or one Shape\_aspect. Each Shape\_aspect defines the applied\_shape for zero, one, or many Width\_dimension objects.

## 5 Application interpreted model

### 5.1 Mapping specification

This clause contains the mapping specification that shows how each UoF and application object of this part of ISO 10303 (see clause 4) maps to one or more AIM constructs (see annex A). Each mapping specifies up to five elements.

**Application element:** The mapping for each application element is specified in a separate subclause below. Application object names are given in title case. Attribute names and assertions are listed after the application object to which they belong and are given in lower case.

**AIM element:** The name of one or more AIM entity data types (see annex A), the term “IDENTICAL MAPPING”, or the term “PATH”. AIM entity data type names are given in lower case. Attributes of AIM entity data types are referred to as <entity name>.<attribute name>. The mapping of an application element may involve more than one AIM element. Each of these AIM elements is presented on a separate line in the mapping specification. The term “IDENTICAL MAPPING” indicates that both application objects involved in an application assertion map to the same instance of an AIM entity data type. The term “PATH” indicates that the application assertion maps to a collection of related AIM entity instances specified by the entire reference path.

**Source:** For those AIM elements that are interpreted from any common resource, this is the ISO standard number and part number in which the resource is defined. For those AIM elements that are created for the purpose of this part of ISO 10303, this is “ISO 10303-“ followed by the number of this part.

Rules: One or more global rules may be specified that apply to the population of the AIM entity data types specified as the AIM element or in the reference path. For rules that are derived from relationships between application objects, the same rule is referred to by the mapping entries of all the involved AIM elements. A reference to a global rule may be accompanied by a reference to the subclause in which the rule is defined.

Reference path: To describe fully the mapping of an application object, it may be necessary to specify a reference path involving several related AIM elements. Each line in the reference path documents the role of an AIM element relative to the AIM element in the line following it. Two or more such related AIM elements define the interpretation of the integrated resources that satisfies the requirement specified by the application object. For each AIM element that has been created for use within this part of ISO 10303, a reference path to its supertype from an integrated resource is specified. For the expression of reference paths and the relationships between AIM elements the following notational conventions apply:

- [] enclosed section constrains multiple AIM elements or sections of the reference path are required to satisfy an information requirement;
- () enclosed section constrains multiple AIM elements or sections of the reference path are identified as alternatives within the mapping to satisfy an information requirement;
- { } enclosed section constrains the reference path to satisfy an information requirement;
- <> enclosed section constrains at one or more required reference path;
- || enclosed section constrains the supertype entity;
- > attribute references the entity or select type given in the following row;
- <- entity or select type is referenced by the attribute in the following row;
- [i] attribute is an aggregation of which a single member is given in the following row;
- [n] attribute is an aggregation of which member n is given in the following row;
- => entity is a supertype of the entity given in the following row;
- <= entity is a subtype of the entity given in the following row;
- = the string, select, or enumeration type is constrained to a choice or value;
- \ the reference path expression continues on the next line;
- \* used in conjunction with braces to indicate that any number of relationship entity data types may be assembled in a relationship tree structure;
- the text following is a comment (normally a clause reference).

## 5.1.1 Casting features UoF

### 5.1.1.1 Casting\_design\_feature

AIM element: casting\_design\_feature  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: casting\_design\_feature<=  
 casting\_feature\_definition  
 [casting\_feature\_definition=>  
 casting\_instanced\_feature]

#### 5.1.1.1.1 draft

AIM element: measure\_representation\_item  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: casting\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'draft'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.1.1.2 machine\_stock

AIM element: machining\_allowance  
Source: ISO 10303-223  
Reference Path: casting\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition=>  
machining\_allowance

### 5.1.1.1.3 casting\_design\_feature to orientation (as placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )  
Reference Path: casting\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'orientation'}  
representation\_item =>  
geometric\_representation\_item =>  
placement

### 5.1.1.2 Casting\_insert

AIM element: casting\_insert  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: casting\_insert<=  
 sand\_cast\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.1.2.1 insert\_name

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: casting\_insert<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'insert name'}  
 representation\_item =>  
 descriptive\_representation\_item

### 5.1.1.2.2 material\_type

AIM element: material\_designation  
Source: ISO 10303-45  
Reference Path: casting\_insert<=  
sand\_cast\_design\_feature<=  
casting\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation

### 5.1.1.2.3 positioning\_method

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: casting\_insert<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'positioning method'}  
representation\_item =>  
descriptive\_representation\_item



### 5.1.1.3 Chaplet

AIM element: chaplet  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: chaplet<=  
 sand\_cast\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.1.3.1 chaplet\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: chaplet<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'chaplet type'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description

### 5.1.1.3.2 chaplet to item\_size (as chaplet\_size)

AIM element: PATH  
Reference Path: chaplet<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'chaplet reference occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'chaplet dimension usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.3.3 chaplet to chaplet\_pad (as pad)

AIM element: PATH  
 Reference Path: chaplet<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'chaplet reference occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'chaplet pad reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 instanced\_feature=>  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 chaplet\_pad

### 5.1.1.4 Chaplet\_pad

AIM element: chaplet\_pad  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: chaplet\_pad<=  
 sand\_cast\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.4.1 chaplet\_pad to item\_size (as pad\_size)

AIM element: PATH  
Reference Path: chaplet\_pad<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'chaplet pad dimension occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'chaplet pad dimension usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.5 Chill

AIM element: chill  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: chill<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.5.1 internal\_or\_external\_chill

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: chill<=  
   sand\_cast\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'internal or external chill'}  
   representation\_item =>  
   descriptive\_representation\_item  
   descriptive\_representation\_item.description  
   {(descriptive\_representation\_item.description = 'true')  
   (descriptive\_representation\_item.description = 'false')}

### 5.1.1.5.2 material\_type

AIM element: material\_designation  
 Source: ISO 10303-45  
 Reference Path: chill<=  
   sand\_cast\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   material\_designation.definitions[i]  
   material\_designation

### 5.1.1.5.3 chill to shape\_element (as chill\_area\_shape)

AIM element: PATH  
Reference Path: chill<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'chill area occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'chill area shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect

### 5.1.1.6 Choke

AIM element: choke  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: choke<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.6.1 chock to item\_size (as chock\_size)

```

AIM element:      PATH
Reference Path:   chock<=
                  gating_design_feature<=
                  casting_feature_definition<=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'chock occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  feature_component_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'chock dimension usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relying_shape_aspect ->
                  shape_aspect =>
                  casting_feature_size

```

### 5.1.1.6.2 choke to location\_element (as vent\_placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: choke<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'location'}  
representation =>  
shape\_representation=>  
location\_shape\_representation

### 5.1.1.7 Connection\_transition

AIM element: representation  
Source: ISO 10303-43

#### 5.1.1.7.1 edge\_round

AIM element: length\_measure\_with\_unit  
Source: ISO 10303-41  
Reference Path: representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'edge round'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}



### 5.1.1.7.2 fillet

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Reference Path: representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'fillet'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.1.8 Core

AIM element: core  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: core<=  
 sand\_cast\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.8.1 core\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: core<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'core type'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {(descriptive\_representation\_item.description = 'baked core')  
 (descriptive\_representation\_item.description = 'cover core')  
 (descriptive\_representation\_item.description = 'isocure')  
 (descriptive\_representation\_item.description = 'ram up core')  
 (descriptive\_representation\_item.description = 'ring core')  
 (descriptive\_representation\_item.description = 'setup core')  
 (descriptive\_representation\_item.description = 'slab core')  
 (descriptive\_representation\_item.description = 'so2 core')}

### 5.1.1.8.2 material\_definition

AIM element: material\_definition  
 Source: ISO 10303-45  
 Reference Path: chill<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 material\_designation.definitions[i]  
 material\_definition  
 {(material\_definition.name = 'ceramic')  
 (material\_definition.name = 'oil bonded sand')  
 (material\_definition.name = 'plaster')  
 (material\_definition.name = 'resin coated sand')}

### 5.1.1.8.3 core to location\_element (as vent\_placement)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: core<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 {representation  
 representation.name = 'vent location'}  
 representation =>  
 shape\_representation=>  
 location\_shape\_representation

#### 5.1.1.8.4 core to sand\_mould (as core\_placement)

AIM element: PATH  
Reference Path: core<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'core occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'core placement reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
sand\_mould

### 5.1.1.8.5 core to vent (as gas\_vent)

AIM element: PATH  
 Reference Path: core<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'core occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='gas vent reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 vent

### 5.1.1.9 Core\_box\_tooling

AIM element: core\_box\_tooling  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: core\_box\_tooling<=  
 casting\_production\_definition<=  
 product\_definition  
 product\_definition.frame\_of\_reference->  
 product\_definition\_context<=  
 application\_context\_element  
 application\_context\_element.name='functional definition'

### 5.1.1.9.1 core\_box\_tooling to production\_core\_box (as core\_box\_equipment)

AIM element: PATH  
Reference Path: core\_box\_tooling<=  
casting\_production\_definition  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'core box tooling occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'production core box reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
casting\_design\_feature=>  
production\_pattern\_definition

### 5.1.1.10 Core\_master

AIM element: core\_master  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: core\_master<=  
core\_box\_tooling<=  
casting\_product\_definition<=  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

**5.1.1.10.1 core\_box\_master\_shrink\_factor**

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: core\_master<=  
                   core\_box\_tooling<=  
                   casting\_product\_definition<=  
                   product\_definition  
                   characterized\_product\_definition=product\_definition  
                   characterized\_product\_definition  
                   characterized\_definition = characterized\_product\_definition  
                   characterized\_definition <-  
                   property\_definition.definition  
                   property\_definition <-  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   {representation =>  
                   shape\_representation =>  
                   shape\_representation\_with\_parameters}  
                   representation  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'core box master shrink factor'}  
                   representation\_item =>  
                   measure\_representation\_item  
                   {measure\_representation\_item <=  
                   measure\_with\_unit =>  
                   length\_measure\_with\_unit

### 5.1.1.11 Core\_print

AIM element: core\_print  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: core\_print<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

#### 5.1.1.11.1 core\_print to numeric\_parameter(as clearance)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: core\_print <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='clearance'  
measure\_representation\_item



**5.1.1.11.2 core\_print to drafted\_surface (as draft)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: core\_print<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'draft'}  
representation =>  
shape\_representation=>  
drafted\_surface

### 5.1.1.11.3 core\_print to item\_size (as core\_print\_dimensions)

AIM element: PATH  
Reference Path: core\_print<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'core print occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'core print size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

#### 5.1.1.11.4 core\_print to shape\_element (as feature)

AIM element: PATH  
 Source: ISO 10303-223  
 Reference Path: core\_print<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'core print occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'core print shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect

#### 5.1.1.12 Ejector\_box

AIM element: ejector\_box  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: ejector\_box<=  
 ejector\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.12.1 ejector\_box to ejector\_plate (as plate\_definition)

AIM element: PATH  
Reference Path: ejector\_box<=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector plate reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
ejector\_design\_feature=>  
ejector\_plate

**5.1.1.12.2 ejector\_box to item\_size (as box\_size)**

```

AIM element:      PATH
Reference Path:   ejector_box<=
                  sand_cast_design_feature<=
                  casting_feature_definition<=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'ejector box occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  feature_component_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'ejector box size usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relating_shape_aspect ->
                  shape_aspect =>
                  casting_feature_size

```

### 5.1.1.12.3 ejector\_box to production\_core\_box (as oriented\_to\_movable\_die)

AIM element: PATH  
Reference Path: ejector\_box<=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector box occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector box die reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
production\_core\_box

#### 5.1.1.12.4 ejector\_box to production\_die\_mould (as oriented\_to\_movable\_die)

AIM element: PATH

Reference Path: ejector\_box<=  
 ejector\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'ejector box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'ejector box die reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 production\_tool=>  
 die\_design\_feature=>  
 production\_die\_mould

### 5.1.1.12.5 ejector\_box to production\_die\_cast\_mould (as oriented\_to\_movable\_die)

AIM element: PATH

Reference Path: ejector\_box<=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector box die reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
production\_tool=>  
die\_design\_feature=>  
production\_die\_cast\_mould



### 5.1.1.12.6 ejector\_box to production\_investment\_cast\_mould (as oriented\_to\_movable\_die)

AIM element: PATH  
 Reference Path: ejector\_box<=  
 ejector\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'ejector box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'ejector box die reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 investment\_design\_feature=>  
 production\_investment\_cast\_mould

### 5.1.1.13 Ejector\_design\_feature

AIM element: ejector\_design\_feature  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: ejector\_design\_feature<=  
 casting\_feature\_definition  
 [casting\_feature\_definition=>  
 casting\_instanced\_feature]

### 5.1.1.13.1 ejector\_design\_feature to orientation (as placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )  
Reference Path: ejector\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'orientation'}  
representation\_item =>  
geometric\_representation\_item =>  
placement

### 5.1.1.14 Ejector\_pins

AIM element: ejector\_pin  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: ejector\_pin<=  
ejector\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.14.1 ejector\_pins to item\_size (as pin\_dimensions)**

```

AIM element:      PATH
Reference Path:   ejector_pin<=
                  ejector_design_feature<=
                  casting_feature_definition =>
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'ejector pin occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  feature_component_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'ejector pin dimension usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relating_shape_aspect ->
                  shape_aspect =>
                  casting_feature_size

```

### 5.1.1.14.2 ejector\_pins to machining\_feature (as thread)

AIM element: PATH  
Reference Path: ejector\_pin<=  
ejector\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector pin occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector pin dimension usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
instanced\_feature=>  
thread

### 5.1.1.14.3 ejector\_pins to pin\_tips (as tip\_definition)

AIM element: PATH  
 Reference Path: ejector\_pin<=  
 ejector\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'ejector pin occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'pin tip reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relating\_shape\_aspect ->  
 shape\_aspect  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 ejector\_design\_feature=>  
 pin\_tip

### 5.1.1.15 Ejector\_plate

AIM element: ejector\_plate  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: ejector\_plate<=  
 ejector\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.15.1 plate\_material

AIM element: material\_designation  
Source: ISO 10303-45  
Reference Path: ejector\_plate<=  
ejector\_design\_feature<=  
casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation

### 5.1.1.15.2 ejector\_plate to item\_size (as plate\_shape)

AIM element: PATH  
Reference Path: ejector\_plate<=  
ejector\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector plate occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector plate dimension usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

**5.1.1.15.3 ejector\_plate to numeric\_parameter (as plate\_height)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: ejector\_plate<=  
 ejector\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'plate height'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

#### 5.1.1.15.4 ejector\_plate to ejector\_pin (as ejector\_pin\_definition)

AIM element: PATH  
Reference Path: ejector\_plate <=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector plate occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector pin reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
ejector\_design\_feature=>  
ejector\_pin



### 5.1.1.15.5 ejector\_plate to reset\_or\_return\_pins (as reset\_or\_return\_pin\_definition)

AIM element: PATH

Reference Path: ejector\_plate<=  
 ejector\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'ejector plate occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'reset or push pin reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relating\_shape\_aspect ->  
 shape\_aspect  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 ejector\_design\_feature=>  
 reset\_or\_push\_pin

### 5.1.1.15.6 ejector\_plate to stop\_pin (as stop\_pin\_definition)

AIM element: PATH  
Reference Path: ejector\_plate<=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector plate occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'stop pin reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
ejector\_design\_feature=>  
stop\_pin

### 5.1.1.16 Ejector\_system

AIM element: ejector\_system  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: ejector\_system<=  
ejector\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.16.1 system\_operated\_by**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path:

```

ejector_system<=
ejector_design_feature<=
casting_feature_definition<=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'system operated by'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'hydraulics')
(descriptive_representation_item.description = 'spring loaded')
(descriptive_representation_item.description = 'simple mechanical')}
  
```

### 5.1.1.16.2 ejector\_system to ejector\_plates (as plate\_definition)

AIM element: PATH  
Reference Path: ejector\_system<=  
ejector\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ejector system occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ejector system reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
ejector\_plate

### 5.1.1.17 Feeder

AIM element: feeder  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: feeder<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.17.1 feeders to connection\_transition (as transition)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: feeder<=  
   gating\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition <=  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation

### 5.1.1.17.2 feeder to item\_size (as parametric\_shape)

AIM element: PATH  
Reference Path: feeder<=  
gating\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'feeder occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'feeder size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

**5.1.1.17.3 feeders to risers (as runner\_connection)**

AIM element: PATH  
 Reference Path: feeder<=  
 gating\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'feeder occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'riser reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relating\_shape\_aspect ->  
 shape\_aspect  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 riser

**5.1.1.18 Filter**

AIM element: filter  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: filter<=  
 gating\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_featuredefinition <=  
 characterized\_object

### 5.1.1.18.1 type\_of\_filter

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: filter<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'type of filter'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description



### 5.1.1.18.2 filter to item\_size (as filter\_shape)

AIM element: PATH  
 Reference Path: filter<=  
   gating\_design\_feature<=  
   casting\_feature\_definition =>  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'filter occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   feature\_component\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description = 'filter size usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relating\_shape\_aspect ->  
   shape\_aspect =>  
   casting\_feature\_size

### 5.1.1.19 Fixturing\_tab

AIM element: fixturing\_tab  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
   subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: fixturing\_tab<=  
   casting\_design\_feature<=  
   {casting\_feature\_definition =>  
   casting\_instanced\_feature}  
   casting\_feature\_definition <=  
   characterized\_object

### 5.1.1.20 Flask

AIM element: flask  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: flask<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

#### 5.1.1.20.1 flask to descriptive\_parameter (as type\_of\_flask)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: flask<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'flask type'}  
representation\_item =>  
descriptive\_representation\_item

**5.1.1.20.2 flask to numeric\_parameter (as cope\_height)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: flask<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'cope height'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.1.20.3 flask to numeric\_parameter (as drag\_height)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: flask<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'drag height'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

#### 5.1.1.20.4 flask to numeric\_parameter (as flask\_length)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: flask<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'flask length'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.1.20.5 flask to numeric\_parameter (as flask width)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: flask<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'flask width'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

**5.1.1.20.6 flask to pin\_center (as pin\_center\_definition)**

```

AIM element:      PATH
Reference Path:   flask<=
                  sand_cast_design_feature
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'pin center reference occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'pin center reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relating_shape_aspect ->
                  shape_aspect
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  sand_cast_design_feature=>
                  pin_center

```

### 5.1.1.20.7 flask to sand\_mould (as generates)

AIM element: PATH  
Reference Path: feeder<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'flask reference occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'sand mould reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
sand\_mould

### 5.1.1.21 Flaskless

AIM element: flaskless  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: flaskless<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object



**5.1.1.21.1 description**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: flaskless<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'description'}  
 representation\_item =>  
 descriptive\_representation\_item

### 5.1.1.21.2 flaskless to mould\_box (as mould\_made\_by)

AIM element: PATH  
Reference Path: flaskless<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'flaskless reference occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'mould box reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
mould\_box

### 5.1.1.22 Gating\_design\_feature

AIM element: gating\_design\_feature  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: gating\_design\_feature<=  
casting\_feature\_definition  
[casting\_feature\_definition=>  
casting\_instanced\_feature]

### 5.1.1.22.1 gating\_design\_feature to orientation (as placement)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )  
 Reference Path: gating\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'orientation'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.1.23 Gating\_system

AIM element: gating\_system  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: gating\_system<=  
gating\_design\_feature<=  
{casting\_feature\_definition=>  
casting\_instanced\_feature}  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition=characterized\_object<-  
property\_definition.definition  
property\_definition  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect  
shape\_aspect.name = 'compound feature gating system'

### 5.1.1.23.1 gating\_system to ingates (as gating\_components)

AIM element: PATH  
 Reference Path: gating\_system<=  
                   gating\_design\_feature<=  
                   casting\_feature\_definition=>  
                   characterized\_object  
                   characterized\_definition=characterized\_object  
                   characterized\_definition<=  
                   property\_definition.definition  
                   property\_definition=>  
                   product\_definition\_shape<=  
                   shape\_aspect.of\_shape  
                   {shape\_aspect=>  
                   composite\_shape\_aspect}  
                   shape\_aspect<=  
                   shape\_aspect\_relationship.relate\_shape\_aspect  
                   {shape\_aspect\_relationship.description='gating system components usage'}  
                   {shape\_aspect\_relationship=>  
                   shape\_defining\_relationship}  
                   shape\_aspect\_relationship  
                   shape\_aspect\_relationship.related\_shape\_aspect->  
                   shape\_aspect=>  
                   casting\_instanced\_feature=>  
                   casting\_feature\_definition=>  
                   gating\_design\_feature=>  
                   ingate

### 5.1.1.23.2 gating\_system to risers (as gating\_components)

AIM element: PATH  
Reference Path: gating\_system<=  
gating\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition<=  
property\_definition.definition  
property\_definition=>  
product\_definition\_shape<=  
shape\_aspect.of\_shape  
{shape\_aspect=>  
composite\_shape\_aspect}  
shape\_aspect<=  
shape\_aspect\_relationship.relate\_shape\_aspect  
{shape\_aspect\_relationship.description='gating system components usage'}  
{shape\_aspect\_relationship=>  
shape\_defining\_relationship}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.related\_shape\_aspect->  
shape\_aspect=>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
riser

### 5.1.1.23.3 gating\_system to runners (as gating\_components))

AIM element: PATH  
 Reference Path: gating\_system<=  
   gating\_design\_feature<=  
   casting\_feature\_definition=>  
   characterized\_object  
   characterized\_definition=characterized\_object  
   characterized\_definition<=  
   property\_definition.definition  
   property\_definition=>  
   product\_definition\_shape<=  
   shape\_aspect.of\_shape  
   {shape\_aspect=>  
   composite\_shape\_aspect}  
   shape\_aspect<=  
   shape\_aspect\_relationship.relate\_shape\_aspect  
   {shape\_aspect\_relationship.description='gating system components usage'}  
   {shape\_aspect\_relationship=>  
   shape\_defining\_relationship}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.related\_shape\_aspect->  
   shape\_aspect=>  
   casting\_instanced\_feature=>  
   casting\_feature\_definition=>  
   gating\_design\_feature=>  
   runner

#### 5.1.1.23.4 gating\_system to sprue (as gating\_components))

AIM element: PATH  
Reference Path: gating\_system<=  
gating\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition<=  
property\_definition.definition  
property\_definition=>  
product\_definition\_shape<=  
shape\_aspect.of\_shape  
{shape\_aspect=>  
composite\_shape\_aspect}  
shape\_aspect<=  
shape\_aspect\_relationship.relate\_shape\_aspect  
{shape\_aspect\_relationship.description='gating system components usage'}  
{shape\_aspect\_relationship=>  
shape\_defining\_relationship}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.related\_shape\_aspect->  
shape\_aspect=>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
sprue

#### 5.1.1.24 Ingates

AIM element: ingate  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: ingate<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object



**5.1.1.24.1 cope\_or\_drag**

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: ingate<=  
gating\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'cope or drag type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
{(descriptive\_representation\_item.description = 'cope')  
(descriptive\_representation\_item.description = 'drag')  
(descriptive\_representation\_item.description = 'cope and drag')}

### 5.1.1.24.2 ingates to ingate\_contacts (as contacts)

AIM element: PATH  
Reference Path: ingate<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ingate occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'ingate contact reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
ingate\_contact

### 5.1.1.24.3 ingates to item\_size (as ingate\_shape)

AIM element: PATH  
 Reference Path: ingate<=  
   gating\_design\_feature  
   casting\_feature\_definition =>  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'ingate occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   feature\_component\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description = 'ingate size usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relate\_shape\_aspect ->  
   shape\_aspect =>  
   casting\_feature\_size

### 5.1.1.25 Ingate\_contacts

AIM element: ingate\_contact  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
   subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: ingate\_contact<=  
   gating\_design\_feature<=  
   {casting\_feature\_definition =>  
   casting\_instanced\_feature}  
   casting\_feature\_definition <=  
   characterized\_object

### 5.1.1.25.1 contact\_placement

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: ingate\_contact<=  
gating\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'cope or drag type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
{(descriptive\_representation\_item.description = 'cope')  
(descriptive\_representation\_item.description = 'drag')  
(descriptive\_representation\_item.description = 'cope and drag')}

**5.1.1.25.2 ingate\_contacts to connection\_transition (as runner\_transition)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: ingate\_contact<=  
   gating\_design\_feature  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition <=  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   {representation.description= 'runner transition'}

### 5.1.1.25.3 ingate\_contacts to connection\_transition (as pattern\_transition)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: ingate\_contact<=  
gating\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
{representation.description= 'pattern transition'}

**5.1.1.25.4 ingate\_contacts to production\_pattern\_definition (as applied\_to)**

AIM element: PATH  
Reference Path: ingate\_contact<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ingate occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n  
'production pattern definition reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
production\_pattern\_definition

### 5.1.1.25.5 ingates\_contacts to runner (as runner\_connection)

AIM element: PATH  
Reference Path: ingate\_contact<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'ingate occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='runner reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
runner



### 5.1.1.25.6 ingate\_contacts to shape\_element (as break\_off\_connection)

AIM element: PATH  
 Reference Path: ingate\_contact<=  
   gating\_design\_feature  
   casting\_feature\_definition <=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'ingate occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   shape\_defining\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description='ingate contact shape usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relate\_shape\_aspect ->  
   shape\_aspect

### 5.1.1.26 Item\_size

AIM element: casting\_feature\_size  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
   subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: casting\_feature\_size<=  
   shape\_aspect

### 5.1.1.26.1 item\_size to numeric\_parameter (as item\_height)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: casting\_feature\_size<=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'height'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

**5.1.1.26.2 item\_size to numeric\_parameter (as item\_length)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: casting\_feature\_size<=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'length'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.1.26.3 item\_size to numeric\_parameter (as item\_width)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: casting\_feature\_size<=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'width'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

**5.1.1.26.4 item\_size to numeric\_parameter (as item\_diameter)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: casting\_feature\_size<=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'diameter'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.1.26.5 **item\_size to numeric\_parameter (as item\_draft)**

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: casting\_feature\_size<=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'draft'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.1.27 **Loose\_piece**

AIM element: loose\_piece  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: loose\_piece<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.27.1 loose\_piece\_material

AIM element: material\_designation.name  
 Source: ISO 10303-45  
 Reference Path: loose\_piece<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 material\_designation.definitions[i]  
 material\_designation  
 material\_designation.name  
 {(material\_designation.name = 'aluminum')  
 (material\_designation.name = 'cast iron')  
 (material\_designation.name = 'urethane')  
 (material\_designation.name = 'wood')}

### 5.1.1.27.2 loose\_piece to shape\_element (as shape)

AIM element: PATH  
 Reference Path: loose\_piece<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'loose piece occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='loose piece shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect

### 5.1.1.28 Master\_pattern

AIM element: master\_pattern  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
          subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: master\_pattern<=  
                  sand\_casting\_tooling<=  
                  casting\_product\_definition<=  
                  product\_definition  
                  product\_definition.frame\_of\_reference->  
                  product\_definition\_context<=  
                  application\_context\_element  
                  application\_context\_element.name='functional definition'



### 5.1.1.28.1 shrink\_factor\_for\_master\_pattern

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: master\_pattern<=  
 sand\_casting\_tooling<=  
 casting\_product\_definition<=  
 product\_definition  
 characterized\_product\_definition=product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'shrink factor for master pattern'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit

### 5.1.1.28.2 master\_pattern to machining\_allowance (as master\_allowance\_on\_master\_pattern)

AIM element: machining\_allowance  
Source: ISO 10303-223  
Reference Path: master\_pattern<=  
sand\_casting\_tooling<=  
casting\_product\_definition<=  
product\_definition  
characterized\_product\_definition=product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
property\_definition=>  
machining\_allowance

### 5.1.1.29 Mould\_box

AIM element: mould\_box  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: mould\_box<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.29.1 mould\_box to descriptive\_parameter (as description)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: mould\_box<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'mould box description'}  
 representation\_item =>  
 descriptive\_representation\_item

### 5.1.1.29.2 mould\_box to item\_size (as box\_size)

AIM element: PATH  
Reference Path: mould\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'mould box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'mould box dimension usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.30 Mould\_locks

AIM element: mould\_locks  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: mould\_locks<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.30.1 mould\_locks to item\_size (as lock\_dimensions)**

```

AIM element:      PATH
Reference Path:   mould_locks<=
                  sand_cast_design_feature<=
                  casting_feature_definition =>
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'mould lock occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  feature_component_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'mould lock usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relying_shape_aspect ->
                  shape_aspect =>
                  casting_feature_size

```

### 5.1.1.30.2 mould\_locks to descriptive\_parameter (as description)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: mould\_locks<=  
sand\_cast\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'mould lock description'}  
representation\_item =>  
descriptive\_representation\_item

### 5.1.1.31 Pattern\_plate

AIM element: pattern\_plate  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: pattern\_plate<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.31.1 type\_of\_plate

AIM element: material\_designation.name  
 Source: ISO 10303-45  
 Reference Path: pattern\_plate <=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition =>  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 material\_designation.definitions[i]  
 material\_designation  
 material\_designation.name  
 material\_designation.name  
 {(material\_designation.name = 'wood')  
 (material\_designation.name = 'metal')  
 (material\_designation.name = 'metal insert')  
 (material\_designation.name = 'wood insert')}

### 5.1.1.31.2 pattern\_plate to item\_size (as pattern\_plate\_dimensions)

AIM element: PATH  
 Reference Path: pattern\_plate<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition =>  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'pattern plate occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'pattern plate usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.1.31.3 **pattern\_plate to location\_element (as mounting\_hole\_locations)**

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: pattern\_plate<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'location'}  
representation =>  
shape\_representation=>  
location\_shape\_representation



### 5.1.1.31.4 pattern\_plat to flask (as defines\_flask\_type)

```

AIM element:      PATH
Reference Path:   pattern_plate<=
                  sand_cast_design_feature<=
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'pattern plate occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description='flask reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relying_shape_aspect ->
                  shape_aspect =>
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  sand_cast_design_feature=>
                  flask

```

### 5.1.1.31.5 pattern\_plate to gating\_system (as gate\_rigging)

```
AIM element:      PATH
Reference Path:   pattern_plate<=
                  sand_cast_design_feature<=
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'pattern plate occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description='gating system reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relate_shape_aspect ->
                  shape_aspect =>
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  gating_design_feature=>
                  gating_system
```

### 5.1.1.31.6 pattern\_plate to mould\_locks (as lock\_definitions)

```

AIM element:      PATH
Reference Path:  pattern_plate<=
                 sand_cast_design_feature<=
                 casting_feature_definition <=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'pattern plate occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 shape_defining_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description='mould lock reference usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relying_shape_aspect ->
                 shape_aspect =>
                 casting_instanced_feature=>
                 casting_feature_definition=>
                 sand_cast_design_feature=>
                 mould_locks

```

### 5.1.1.31.7 **pattern\_plate to parting\_surface (as parting\_line)**

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: pattern\_plate<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'parting line'}  
representation =>  
shape\_representation=>  
parting\_surface

**5.1.1.31.8 pattern\_plate to pin\_center (as pin\_center\_definition)**

```

AIM element:      PATH
Reference Path:   pattern_plate<=
                  sand_cast_design_feature<=
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'pattern plate occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description='pin center reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relating_shape_aspect ->
                  shape_aspect =>
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  sand_cast_design_feature=>
                  pin_center

```

### 5.1.1.31.9 pattern\_plate to moulding\_equipment (as moulding\_machine)

AIM element: PATH  
Reference Path: pattern\_plate<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
action\_method\_item= shape\_aspect  
action\_method\_item<-  
applied\_action\_method\_assignment.items[i]  
action\_method\_assignment  
action\_method\_assignment.assigned\_action\_method  
action\_method  
supported\_item=action\_method  
supported\_item<-  
action\_resource.usage[i]  
action\_resource=>  
machine=>  
casting\_equipment=>  
moulding\_equipment

### 5.1.1.32 Pin\_center

AIM element: pin\_center  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: pin\_center<=  
sand\_cast\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.32.1 pin\_center to numeric\_parameter\_with\_tolerance (as dimension)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: pin\_center<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'dimension'}  
 representation\_item =>  
 [measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}]  
 [qualified\_representation\_item]

### 5.1.1.33 Pin\_tips

AIM element: pin\_tip  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: pin\_tip<=  
 ejector\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.33.1 pin\_tips to shape\_element (as shape)

AIM element: PATH  
Reference Path: pin\_tip<=  
ejector\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'pin tip occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='pin tip shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relatng\_shape\_aspect ->  
shape\_aspect

### 5.1.1.34 General\_rib

AIM element: rib  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference path: rib<=  
casting\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object  
{characterized\_object  
characterized\_object.description = 'general'}



**5.1.1.34.1 general\_rib to numeric\_parameter (as part\_rib\_fillet)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )

Reference path: rib <=

```

casting_design_feature
casting_feature_definition<=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'part rib fillet'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

```

### 5.1.1.34.2 **general\_rib to numeric\_parameter (as wall\_to\_floor\_fillet)**

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'wall to floor fillet'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.1.34.3 general\_rib to numeric\_parameter (as wall\_to\_top\_radius)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14 )

Reference path: rib <=

```

casting_design_feature
casting_feature_definition<=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'wall to top radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

```

#### 5.1.1.34.4 **general\_rib to face\_shape\_element(as walls\_top\_floor)**

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.name = 'walls top floor'}  
representation =>  
shape\_representation=>  
face\_shape\_representation

### 5.1.1.34.5 general\_rib to planar\_element(as walls\_top\_floor)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference path: rib <=  
 casting\_design\_feature  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.name = 'walls top floor'}  
 representation =>  
 shape\_representation=>  
 planar\_shape\_representation

### 5.1.1.35 Planar\_rib

AIM element: rib  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference path: rib<=  
 casting\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object  
 {characterized\_object  
 characterized\_object.description = 'planar'}

### 5.1.1.35.1 Planar\_rib to numeric\_parameter (as part\_rib\_radius)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'part rib radius'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

**5.1.1.35.2 Planar\_rib to numeric\_parameter (as rib\_extent\_fillet)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference path: rib <=

```

casting_design_feature
casting_feature_definition<=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'rib extent fillet'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

```

### 5.1.1.35.3 Planar\_rib to numeric\_parameter (as transition\_to\_top\_radius)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'transition to top radius'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}



#### 5.1.1.35.4 Planar\_rib to numeric\_parameter (as wall\_to\_top\_radius)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference path: rib <=

```

casting_design_feature
casting_feature_definition<=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'wall to top radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

```

### 5.1.1.35.5 Planar\_rib to numeric\_parameter (as wall\_to\_floor\_fillet)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'wall to floor fillet'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

**5.1.1.35.6 Planar\_rib to face\_shape\_element(as top)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference path: rib <=  
   casting\_design\_feature  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition <=  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.name = 'walls top floor'}  
   representation =>  
   shape\_representation=>  
   face\_shape\_representation

### 5.1.1.35.7 Planar\_rib to planar\_element(as top)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: rib <=  
casting\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.name = 'walls top floor'}  
representation =>  
shape\_representation=>  
planar\_shape\_representation

### 5.1.1.35.8 Planar\_rib to planar\_element(as walls)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference path: rib <=  
 casting\_design\_feature  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.name = 'walls top floor'}  
 representation =>  
 shape\_representation=>  
 planar\_shape\_representation

### 5.1.1.36 Production\_core\_box

AIM element: production\_core\_box  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: production\_core\_box<=  
 casting\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.36.1 box\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'core box type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.1.36.2 vent\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_core\_box<=  
   sand\_cast\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'core vent type'}  
   representation\_item =>  
   descriptive\_representation\_item  
   descriptive\_representation\_item.description

### 5.1.1.36.3 core\_box\_material

AIM element: material\_designation.name  
Source: ISO 10303-45  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
material\_designation.definitions[i]  
material\_designation  
material\_designation.name  
{(material\_designation.name = 'plaster')  
(material\_designation.name = 'expandable polystyrene')  
(material\_designation.name = 'iron')  
(material\_designation.name = 'soft wood')  
(material\_designation.name = 'shell')  
(material\_designation.name = 'aluminum')  
(material\_designation.name = 'epoxy')  
(material\_designation.name = 'wood metal reinforced')  
(material\_designation.name = 'brass')  
(material\_designation.name = 'hard wood')  
(material\_designation.name = 'urethane')  
(material\_designation.name = 'other')  
(material\_designation.name = 'steel')  
(material\_designation.name = 'ceramic')  
(material\_designation.name = 'polymethylmethacrylate')  
(material\_designation.name = 'plywood')} }



### 5.1.1.36.4 number\_of\_impressions

AIM element: count\_measure  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: production\_core\_box<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'number of impressions'}  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

### 5.1.1.36.5 shrink\_factor

AIM element: length\_measure\_with\_unit  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'shrink factor'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit

**5.1.1.36.6 production\_core\_box to casting\_insert (as inserts)**

```

AIM element:      PATH
Reference Path:   production_core_box<=
                  sand_cast_design_feature<=
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'production core box occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description='casting insert reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relying_shape_aspect ->
                  shape_aspect =>
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  sand_cast_design_feature=>
                  casting_insert

```

### 5.1.1.36.7 production\_core\_box to chaplet\_pad (as chaplet\_pad\_definition)

AIM element: PATH  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production core box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='chaplet pad reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
chaplet\_pad

**5.1.1.36.8 production\_core\_box to core (as core\_definiton)**

```

AIM element:      PATH
Reference Path:  production_core_box<=
                 sand_cast_design_feature<=
                 casting_feature_definition <=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'production core box occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 shape_defining_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description='core reference usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relateing_shape_aspect ->
                 shape_aspect =>
                 casting_instanced_feature=>
                 casting_feature_definition=>
                 sand_cast_design_feature=>
                 core

```

### 5.1.1.36.9 production\_core\_box to core\_equipment (as core\_made\_by)

AIM element: PATH  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
action\_method\_item=shape\_aspect  
action\_method\_item<-  
applied\_action\_method\_assignment.items[i]  
action\_method\_assignment  
action\_method\_assignment.assigned\_action\_method  
action\_method  
supported\_item=action\_method  
supported\_item<-  
action\_resource.usage[i]  
action\_resource=>  
machine=>  
casting\_equipment=>  
core\_equipment

**5.1.1.36.10 production\_core\_box to core\_print (as core\_print\_definition)**

```

AIM element:      PATH
Reference Path:  production_core_box<=
                 sand_cast_design_feature<=
                 casting_feature_definition <=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'production core box occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 shape_defining_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description='core print reference usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relate_shape_aspect ->
                 shape_aspect =>
                 casting_instanced_feature=>
                 casting_feature_definition=>
                 sand_cast_design_feature=>
                 core_print

```

### 5.1.1.36.11 production\_core\_box to drafted\_surface (as draft)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
drafted\_surface



**5.1.1.36.12 production\_core\_box to loose\_piece(as loose\_piece\_requirement)**

AIM element: PATH  
 Reference Path: production\_core\_box<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production core box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='loose piece reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 loose\_piece

### 5.1.1.36.13 production\_core\_box to in\_facility\_location (as core\_box\_location)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
in\_facility\_location

### 5.1.1.36.14 production\_core\_box to person\_in\_organization (as core\_box\_location)

AIM element: PATH  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_instanced\_feature<=  
shape\_aspect<=  
applied\_person\_and\_organization\_assignment.items  
applied\_person\_and\_organization\_assignment<=  
person\_and\_organization\_assignment  
{person\_and\_organization\_assignment.role->  
person\_and\_organization\_role  
person\_and\_organization\_role.name='core box stored'}  
person\_and\_organization\_assignment.assigned\_person\_and\_organization->  
person\_and\_organization

**5.1.1.36.15 production\_core\_box to item\_size (as guide\_pin)**

AIM element: PATH

Reference Path: production\_core\_box<=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition =>  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production core box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'production core box usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.1.36.16 production\_core\_box to item\_size (as outside\_shape\_of\_core\_box)

AIM element: PATH  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production core box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'outside shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.36.17 production\_core\_box to shape\_aspect (as outside\_shape\_of\_core\_box)

AIM element: PATH  
 Source: ISO 10303-223  
 Reference Path: production\_core\_box <=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production core box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='outside shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect

### 5.1.1.36.18 production\_core\_box to parting\_surface (as cope\_and\_drag\_parting)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
{representation.name= 'cope and drag parting'}  
parting\_surface

### 5.1.1.36.19 production\_core\_box to parting\_surface (as offset\_parting\_surface)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: production\_core\_box<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 shape\_representation=>  
 {representation.name= 'offset parting'}  
 parting\_surface

### 5.1.1.36.20 production\_core\_box to planar\_element(as vent\_placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
planar\_shape\_representation



### 5.1.1.36.21 production\_core\_box to shape\_element (as investment\_area\_shape)

AIM element: PATH  
 Source: ISO 10303-223  
 Reference Path: production\_core\_box <=  
 sand\_cast\_design\_feature  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production core box occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='investment area shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect

### 5.1.1.36.22 production\_core\_box to shape\_element (as impression\_shape)

AIM element: PATH  
Source: ISO 10303-223  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production core box occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='impression shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect

### 5.1.1.36.23 production\_core\_box to machining\_allowance (as machine\_stock\_on\_tool)

AIM element: machining\_allowance  
Source: ISO 10303-223  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
property\_definition=>  
machining\_allowance

### 5.1.1.37 Production\_pattern\_definition

AIM element: production\_pattern\_definition  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature <=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.1.37.1 condition

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature <=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'condition'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {(descriptive\_representation\_item.description = 'good')  
 (descriptive\_representation\_item.description = 'bad')  
 (descriptive\_representation\_item.description = 'fair')  
 (descriptive\_representation\_item.description = 'new')}

### 5.1.1.37.2 number\_of\_patterns\_mounted

AIM element: count\_measure  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'number of patterns mounted'}  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

**5.1.1.37.3 shrink\_factor**

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_pattern\_definition<=  
   sand\_cast\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'shrink factor'}  
   representation\_item =>  
   measure\_representation\_item  
   {measure\_representation\_item <=  
   measure\_with\_unit =>  
   length\_measure\_with\_unit}

#### 5.1.1.37.4 production\_pattern\_definition to casting\_insert (as inserts)

AIM element: PATH  
Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='casting insert reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
casting insert

### 5.1.1.37.5 production\_pattern\_definition to chaplet\_pad (as chaplet\_pad\_definition)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production pattern definition occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='chaplet pad reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 chaplet\_pad

### 5.1.1.37.6 production\_pattern\_definition to chill (as chill\_pattern)

AIM element: PATH  
Reference Path: production\_pattern\_definition<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='chill reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
chill



### 5.1.1.37.7 production\_pattern\_definition to in\_facility\_location (as pattern\_location)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: production\_pattern\_definition<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 in\_facility\_location

### 5.1.1.37.8 production\_pattern\_definition to production\_core\_box(as cores\_for\_pattern)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='production core box reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
production\_core\_box

### 5.1.1.37.9 production\_pattern\_definition to core\_print (as core\_print\_definition)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production pattern definition occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='core print reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 core\_print

### 5.1.1.37.10 production\_pattern\_definition to loose\_piece (as loose\_piece\_requirement)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='loose piece reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
loose\_piece

### 5.1.1.37.11 production\_pattern\_definition to parting\_surface (as cope\_and\_drag\_parting)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 (representation.name='cope and drag parting')  
 shape\_representation=>  
 parting\_surface

### 5.1.1.37.12 production\_pattern\_definition to parting\_surface (as offset\_parting\_surface)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
(representation.name='offset parting surface')  
parting\_surface

### 5.1.1.37.13 production\_pattern\_definition to pattern\_plate (as match\_plate\_patterns)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production pattern definition occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='pattern plate reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 pattern\_plate

### 5.1.1.37.14 production\_pattern\_definition to riser (as riser\_on\_pattern)

AIM element: PATH  
Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='riser reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
riser



### 5.1.1.37.15 production\_pattern\_definition to riser\_contact (as riser\_on\_pattern)

AIM element: PATH

Reference Path: production\_pattern\_definition <=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production pattern definition occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='riser contact reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 riser\_contact

### 5.1.1.37.16 production\_pattern\_definition to shape\_element (as invest\_area)

AIM element: PATH  
Source: ISO 10303-223  
Reference Path: production\_pattern\_definition <=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production pattern definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='production pattern definition shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect

### 5.1.1.38 Reset\_or\_return\_pins

AIM element: reset\_or\_push\_pin  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: reset\_or\_return\_pin<=  
ejector\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.38.1 spring\_description

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: reset\_or\_push\_pin<=  
                   ejector\_design\_feature<=  
                   casting\_feature\_definition<=  
                   characterized\_object  
                   characterized\_definition = characterized\_object  
                   characterized\_definition <-  
                   property\_definition.definition  
                   property\_definition <-  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   {representation =>  
                   shape\_representation =>  
                   shape\_representation\_with\_parameters}  
                   representation  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'spring description'}  
                   representation\_item =>  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

### 5.1.1.38.2 reset\_or\_return\_pins to item\_size (as pin\_dimensions)

AIM element: PATH  
Reference Path: reset\_or\_return\_pins<=  
ejector\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'reset or push pin occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'reset or push pin dimensions size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.39 Rib

AIM element: rib  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference path: rib<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object  
{characterized\_object  
(characterized\_object.description = 'general')  
(characterized\_object.description = 'planar')}

### 5.1.1.40 Risers

AIM element: riser  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
          subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: riser<=  
                  gating\_design\_feature<=  
                  {casting\_feature\_definition =>  
                  casting\_instanced\_feature}  
                  casting\_feature\_definition <=  
                  characterized\_object

### 5.1.1.40.1 riser\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: riser<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'riser type';  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
{(descriptive\_representation\_item.description = 'open')  
(descriptive\_representation\_item.description = 'blind')  
(descriptive\_representation\_item.description = 'open\_with\_radiation\_sheld')  
(descriptive\_representation\_item.description = 'top')  
(descriptive\_representation\_item.description = 'top\_with\_radiation\_sheld')}}

**5.1.1.40.2 riser\_sleeve**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: riser<=  
   gating\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'riser sleeve'}  
   representation\_item =>  
   descriptive\_representation\_item  
   descriptive\_representation\_item.description  
   {(descriptive\_representation\_item.description = 'insulating')  
   (descriptive\_representation\_item.description = 'exothermic')  
   (descriptive\_representation\_item.description = 'exothermic and insulating')}

### 5.1.1.40.3 risers to item\_size (as riser\_dimensions)

AIM element: PATH  
Reference Path: riser<=  
gating\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'riser occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'riser size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size



**5.1.1.40.4 risers to item\_size (as riser\_neck)**

AIM element: PATH  
 Reference Path: riser<=  
   gating\_design\_feature  
   casting\_feature\_definition =>  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'riser occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   feature\_component\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description = 'riser neck size usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relying\_shape\_aspect ->  
   shape\_aspect =>  
   casting\_feature\_size

### 5.1.1.40.5 risers to riser\_contact (as contact\_definition)

AIM element: PATH  
Reference Path: riser<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'riser occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='riser contact reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
riser\_contact

**5.1.1.40.6 risers to well (as riser\_well)**

AIM element: PATH  
Reference Path: riser<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'riser occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='well reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
well

### 5.1.1.40.7 risers to item\_size (as riser\_core)

AIM element: PATH  
Reference Path: riser<=  
gating\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'riser occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'riser core size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.41 Riser\_contact

AIM element: riser\_contact  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: riser\_contact<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.41.1 contact\_placement**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: riser\_contact<=  
 gating\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'cope pr drag'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {(descriptive\_representation\_item.description = 'cope')  
 (descriptive\_representation\_item.description = 'drag')  
 (descriptive\_representation\_item.description = 'cope and drag')}

### 5.1.1.41.2 riser\_contact to shape\_element (as break\_off\_connection)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: riser\_contact<=  
ejector\_design\_feature  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'riser contact occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='riser contact shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect

### 5.1.1.42 Runners

AIM element: runner  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: runner<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.42.1 runner\_placement

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: runner<=  
   gating\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'cope or drag type'}  
   representation\_item =>  
   descriptive\_representation\_item  
   descriptive\_representation\_item.description  
   {(descriptive\_representation\_item.description = 'cope')  
   (descriptive\_representation\_item.description = 'drag')  
   (descriptive\_representation\_item.description = 'cope and drag')}

### 5.1.1.42.2 runners to feeders (as runner\_connection)

AIM element: PATH  
Reference Path: runner<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'runner occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='feeder reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
feeder



**5.1.1.42.3 runners to item\_size (as runner\_dimensions)**

```

AIM element:      PATH
Reference Path:  runner<=
                 gating_design_feature<=
                 casting_feature_definition =>
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'runner occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 feature_component_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description = 'runner dimensions size usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relating_shape_aspect ->
                 shape_aspect =>
                 casting_feature_size

```

#### 5.1.1.42.4 runners to choke (as runner\_choke)

AIM element: PATH  
Reference Path: runner<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'runner occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='choke reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relatng\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
choke

### 5.1.1.42.5 runners to filter (as metal\_flow\_filter)

AIM element: PATH  
 Reference Path: runner<=  
 gating\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'runner occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='filter reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 filter

### 5.1.1.42.6 runners to sprues (as sprue\_connections)

AIM element: PATH  
Reference Path: runner<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'runner occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='sprue reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
sprues

### 5.1.1.42.7 runners to vent (as runner\_vent)

AIM element: PATH  
 Reference Path: runner<=  
   gating\_design\_feature<=  
   casting\_feature\_definition <=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'runner occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   shape\_defining\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description='vent reference usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relating\_shape\_aspect ->  
   shape\_aspect =>  
   casting\_instanced\_feature=>  
   casting\_feature\_definition=>  
   gating\_design\_feature=>  
   vent

### 5.1.1.43 Sand\_cast\_design\_feature

AIM element: sand\_cast\_design\_feature  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
   subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: sand\_cast\_design\_feature<=  
   casting\_feature\_definition  
   [casting\_feature\_definition=>  
   casting\_instanced\_feature]

### 5.1.1.43.1 Sand\_cast\_design\_feature to orientation (as placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'orientation'}  
representation\_item =>  
geometric\_representation\_item =>  
placement

### 5.1.1.44 Sand\_casting\_tooling

AIM element: sand\_cast\_tooling  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: sand\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

**5.1.1.44.1 pattern\_number**

AIM element: product\_definition.id  
 Source: ISO 10303-41  
 Reference Path: sand\_cast\_tooling<=  
                   casting\_production\_definition  
                   product\_definition  
                   product\_definition.formation->  
                   product\_definition\_formation  
                   product\_definition\_formation.id

**5.1.1.44.2 sand\_casting\_tooling to material (as material\_definition)**

AIM element: PATH  
 Reference Path: sand\_cast\_tooling<=  
                   casting\_production\_definition  
                   product\_definition  
                   product\_definition\_relationship.relying\_product\_definition  
                   {product\_definition\_relationship.description='raw material'  
                   product\_definition\_relationship =>  
                   product\_definition\_usage =>  
                   make\_from\_usage\_option}  
                   product\_definition\_relationship  
                   product\_definition\_relationship.related\_product\_definition ->  
                   product\_definition  
                   {product\_definition  
                   characterized\_product\_definition = product\_definition  
                   characterized\_product\_definition  
                   characterized\_definition = characterized\_product\_definition  
                   characterized\_definition <-  
                   material\_designation.definitions[i]  
                   material\_designation}

### 5.1.1.44.3 sand\_casting\_tooling to material (as tooling\_material)

AIM element: material\_designation.name  
Source: ISO 10303-45  
Reference Path: sand\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
product\_definition\_relationship.relateing\_product\_definition  
{product\_definition\_relationship.description='raw material'  
product\_definition\_relationship =>  
product\_definition\_usage =>  
make\_from\_usage\_option}  
product\_definition\_relationship  
product\_definition\_relationship.related\_product\_definition ->  
product\_definition  
{product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
material\_designation.definitions[i]  
material\_designation}  
material\_designation.name  
{(material\_designation.name = 'aluminum')  
(material\_designation.name = 'cast iron')  
(material\_designation.name = 'urethane')  
(material\_designation.name = 'wood')}



#### 5.1.1.44.4 sand\_casting\_tooling to production\_pattern\_definition (as pattern\_equipment)

AIM element: PATH  
 Reference Path: sand\_cast\_tooling<=  
 casting\_production\_definition  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'sand casting tooling occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description=\n  
 'production pattern definition reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 casting\_design\_feature=>  
 production\_pattern\_definition

#### 5.1.1.45 Sand\_mould

AIM element: sand\_mould  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: sand\_mould<=  
 Sand\_cast\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.45.1 sand\_mould\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: sand\_mould<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'sand mod characteristics'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
{(descriptive\_representation\_item.description = 'green sand material')  
(descriptive\_representation\_item.description = 'skin dried method')  
(descriptive\_representation\_item.description = 'cold box mould method')  
(descriptive\_representation\_item.description = 'no bake mould method')}

**5.1.1.45.2 sand\_mould to chaplet (as chaplet\_definition)**

```

AIM element:      PATH
Reference Path:  sand_mould<=
                 sand_cast_design_feature<=
                 casting_feature_definition <=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'sand mould occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 shape_defining_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description='chaplet reference usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relying_shape_aspect ->
                 shape_aspect =>
                 casting_instanced_feature=>
                 casting_feature_definition=>
                 sand_cast_design_feature=>
                 chaplet

```

### 5.1.1.45.3 sand\_mould to chill (as chill\_definition)

AIM element: PATH  
Reference Path: sand\_mould<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sand mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='chill reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
chill

#### 5.1.1.45.4 sand\_mould to equipment (as mould\_made\_by)

AIM element: PATH  
 Reference Path: sand\_mould<=  
 sand\_cast\_design\_feature<=  
 casting\_feature\_definition=>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 action\_method\_item= shape\_aspect  
 action\_method\_item<-  
 applied\_action\_method\_assignment.items[i]  
 action\_method\_assignment  
 action\_method\_assignment.assigned\_action\_method  
 action\_method  
 supported\_item=action\_method  
 supported\_item<-  
 action\_resource.usage[i]  
 action\_resource=>  
 machine=>  
 casting\_equipment=>

#### 5.1.1.46 Sprue

AIM element: sprue  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: sprue<=  
 gating\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.46.1 spur\_core

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: sprue<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'sprue core'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

**5.1.1.46.2 sprues to item\_size (as sprue\_dimensions)**

```

AIM element:    PATH
Reference Path: sprue<=
                gating_design_feature<=
                casting_feature_definition =>
                characterized_object
                characterized_definition = characterized_object
                characterized_definition <-
                property_definition.definition
                property_definition =>
                product_definition_shape <-
                shape_aspect.of_shape
                {shape_aspect
                shape_aspect.description = 'sprue occurrence'}
                shape_aspect <-
                shape_aspect_relationship.related_shape_aspect
                {[shape_aspect_relationship =>
                feature_component_relationship]
                [shape_aspect_relationship
                shape_aspect_relationship.description = 'sprue size usage']}
                shape_aspect_relationship
                shape_aspect_relationship.relating_shape_aspect ->
                shape_aspect =>
                casting_feature_size

```

### 5.1.1.46.3 sprues to filter (as metal\_flow\_filter)

AIM element: PATH  
Reference Path: sprue<=  
gating\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='filter reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
filter



**5.1.1.46.4 sprues to well (as sprue\_well)**

```

AIM element:      PATH
Reference Path:   sprue<=
                  gating_design_feature<=
                  casting_feature_definition <=
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'sprue occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  shape_defining_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description='well reference usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relate_shape_aspect ->
                  shape_aspect =>
                  casting_instanced_feature=>
                  casting_feature_definition=>
                  gating_design_feature=>
                  well

```

### 5.1.1.46.5 sprues to item\_size (as pouring\_cup

AIM element: PATH  
Reference Path: sprue<=  
gating\_design\_feature  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'sprue pouring cup size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.1.47 Stop\_pin

AIM element: stop\_pin  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: stop\_pin<=  
ejector\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.1.47.1 stop\_pin to item\_size (as size)

AIM element: PATH  
 Reference Path: stop\_pin<=  
 ejector\_design\_feature  
 casting\_feature\_definition =>  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'stop pin occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'stop pin size usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.1.48 Vent

AIM element: vent  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: vent<=  
 gating\_design\_feature  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.1.48.1 type\_of\_vent

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: vent<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'type of vent'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
(descriptive\_representation\_item.description= 'air')  
(descriptive\_representation\_item.description= 'gas')

**5.1.1.48.2 vent to item\_size (as chock\_vent\_size)**

AIM element: PATH  
 Reference Path: vent<=  
 gating\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'vent occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'vent reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.1.48.3 vent to location\_element (as vent\_placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: vent<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'location'}  
representation =>  
shape\_representation=>  
location\_shape\_representation

### 5.1.1.49 Well

AIM element: well  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: well<=  
gating\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

**5.1.1.49.1 well to item\_size (as well\_dimensions)**

```

AIM element:      PATH
Reference Path:   well<=
                  gating_design_feature
                  casting_feature_definition =>
                  characterized_object
                  characterized_definition = characterized_object
                  characterized_definition <-
                  property_definition.definition
                  property_definition =>
                  product_definition_shape <-
                  shape_aspect.of_shape
                  {shape_aspect
                  shape_aspect.description = 'well occurrence'}
                  shape_aspect <-
                  shape_aspect_relationship.related_shape_aspect
                  {[shape_aspect_relationship =>
                  feature_component_relationship]
                  [shape_aspect_relationship
                  shape_aspect_relationship.description = 'well size usage']}
                  shape_aspect_relationship
                  shape_aspect_relationship.relating_shape_aspect ->
                  shape_aspect =>
                  casting_feature_size

```

### 5.1.1.49.2 well to parting\_surface (as drag\_surface)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: well<=  
gating\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
parting\_surface

## 5.1.2 Casting\_prerequisites UoF

### 5.1.2.1 Casting\_service\_data

AIM element: casting\_service\_data  
Source: ISO 10303-223  
Reference Path: casting\_service\_data<=  
customer\_casting\_requirement<=  
property\_definition



### 5.1.2.1.1 casting\_service\_data to corrosion\_service (as corrosion\_service\_requirement)

AIM element: PATH  
 Reference Path: casting\_service\_data<=  
 customer\_casting\_requirement<=  
 property\_definition<-  
 property\_definition\_relationship.relating\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 corrosion service

### 5.1.2.1.2 casting\_service\_data to descriptive\_parameter(as comments)

AIM element: PATH  
 Reference Path: casting\_service\_data<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='comments'  
 descriptive\_representation\_item

### 5.1.2.1.3 wear\_with\_lubrication

AIM element: descriptive\_representation\_item  
Source: ISO 10303-45  
Reference Path: casting\_service\_data<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='wear or abrasion lubrication'  
descriptive\_representation\_item  
descriptive\_representation\_item.description='good'  
descriptive\_representation\_item.description='intermittent'  
descriptive\_representation\_item.description='none'

### 5.1.2.1.4 casting\_service\_data to elevated\_temperature\_service (as elevated\_temperature\_service)

AIM element: PATH  
Reference Path: casting\_service\_data<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relying\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
elevated temperature service

### 5.1.2.1.5 casting\_service\_data to finishing\_or\_machining\_operations (as machining\_and\_finishing\_requirements)

AIM element: PATH  
Reference Path: casting\_service\_data<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relying\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
finishing and machining operations

**5.1.2.1.6 casting\_service\_data to material (as subject\_to\_abrasive\_wear\_by)**

AIM element: PATH  
 Reference Path: casting\_service\_data<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='comments'  
 descriptive\_representation\_item  
 descriptive\_representation\_item.name='subject to abrasize wear against'

**5.1.2.1.7 casting\_service\_data to material (as subject\_to\_wear\_against)**

AIM element: PATH  
 Reference Path: casting\_service\_data<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='comments'  
 descriptive\_representation\_item  
 descriptive\_representation\_item.name='subject to wear against'

### 5.1.2.1.8 casting\_service\_data to mechanical\_stress (as mechanical\_stress\_requirements)

AIM element: PATH  
Reference Path: casting\_service\_data<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
mechanical\_stress

### 5.1.2.2 Corrosion\_Service

AIM element: corrosion\_service  
Source: ISO 10303-223  
Reference Path: corrosion\_service<=  
property\_definition

#### 5.1.2.2.1 corrosion\_service to descriptive\_parameter(as corrosive\_environment)

AIM element: PATH  
Reference Path: corrosion\_service<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='corrosive environment'  
descriptive\_representation\_item

### 5.1.2.2.2 corrosion\_service to material (as corrosive\_material)

AIM element: PATH  
 Reference Path: corrosion\_service<=  
                   property\_definition  
                   represented\_definition=property\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation.used\_representation->  
                   representation  
                   representation.items[i] ->  
                   representation\_item  
                   representation\_item.name='comments'  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.name='corrosive material'

### 5.1.2.2.3 corrosion\_service to numeric\_parameter(as ph value)

AIM element: PATH  
 Reference Path: corrosion\_service<=  
                   property\_definition  
                   represented\_definition=property\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation.used\_representation->  
                   representation  
                   representation.items[i] ->  
                   representation\_item  
                   representation\_item.name='ph value'  
                   measure\_representation\_item

#### 5.1.2.2.4 corrosion\_service to numeric\_parameter(as temperature)

AIM element: PATH  
Reference Path: corrosion\_service<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='temperature'  
measure\_representation\_item

#### 5.1.2.3 Customer\_casting\_requirement

AIM element: customer\_casting\_requirement  
Source: ISO 10303-223  
Reference Path: customer\_casting\_requirement<=  
property\_definition

##### 5.1.2.3.1 description

AIM element: property\_definition.description  
Source: ISO 10303-41  
Reference Path: customer\_casting\_requirement<=  
property\_definition  
property\_definition.description

### 5.1.2.3.2 new\_tooling\_required

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'new tooling required'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'customers tool shop')  
 (descriptive\_representation\_item.description = 'foundry tool shop')}

### 5.1.2.3.3 specific\_comments

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'specific comments'}  
 representation\_item =>  
 descriptive\_representation\_item

#### 5.1.2.3.4 customer\_casting\_requirement to illustration (as illustration)

AIM element: PATH  
Reference Path: customer\_casting\_requirement<=  
property\_definition  
document\_reference\_item=property\_definition  
document\_reference\_item <-  
applied\_document\_reference.items [i]  
applied\_document\_reference <=  
document\_reference  
document\_reference.assigned\_document ->  
document=>  
illustration

#### 5.1.2.3.5 customer\_casting\_requirement to shape\_aspect (as applies\_to)

AIM element: PATH  
Reference Path: customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relatng\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition  
property\_definition.definition->  
shape\_aspect

#### 5.1.2.3.6 customer\_casting\_requirement to specification (as referenced\_from)

AIM element: PATH  
Reference Path: customer\_casting\_requirement<=  
property\_definition  
document\_reference\_item =property\_definition  
document\_reference\_item <-  
applied\_document\_reference.items[i]  
applied\_document\_reference <=  
document\_reference  
document\_reference.assigned\_document ->  
document  
{document.kind->  
document\_type  
document\_type.product\_data\_type= 'customer casting requirement'}



### 5.1.2.4 Elevated\_Temperature\_Service

AIM element: elevated\_temperature\_service  
 Source: ISO 10303-223  
 Reference Path: elevated\_temperature\_service<=  
 property\_definition

#### 5.1.2.4.1 elevated\_temperature\_service to descriptive\_parameter(as service\_descriptions)

AIM element: PATH  
 Reference Path: elevated\_temperature\_service<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='service descriptions'  
 descriptive\_representation\_item

#### 5.1.2.4.2 elevated\_temperature\_service to numeric\_parameter(as maximum\_temperature)

AIM element: PATH  
 Reference Path: elevated\_temperature\_service<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='maximum temperature'  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.2.4.3 elevated\_temperature\_service to numeric\_parameter(as minimum\_cyclic\_temperature)

AIM element: PATH  
Reference Path: elevated\_temperature\_service<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='minimum cyclic temperature'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.2.4.4 elevated\_temperature\_service to numeric\_parameter(as maximum\_cyclic\_temperature)

AIM element: PATH  
Reference Path: elevated\_temperature\_service<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='maximum cyclic temperature'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.2.5 Finishing\_or\_machining\_operations

AIM element: finishing\_or\_machining\_operations  
 Source: ISO 10303-223  
 Reference Path: finishing\_or\_machining\_operations<=  
 property\_definition

#### 5.1.2.5.1 finishing\_or\_machining\_operations\_to\_descriptive\_parameter(as comments)

AIM element: PATH  
 Reference Path: finishing\_or\_machining\_operations<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='comments'  
 descriptive\_representation\_item

#### 5.1.2.5.2 finishing\_or\_machining\_operations\_to\_machining\_activity (as finishing\_operations)

AIM element: PATH  
 Reference Path: finishing\_or\_machining\_operations<=  
 property\_definition  
 action\_method\_item=property\_definition  
 action\_method\_item<-  
 applied\_action\_method\_assignment.items[i]  
 applied\_action\_method\_assignment<=  
 action\_method\_assignment  
 {action\_method\_assignment.role->  
 action\_method\_role.name='finishing operation'}  
 action\_method\_assignment.assigned\_action\_method->  
 action\_method=>  
 process\_plan\_activity=>  
 non\_casting\_operation=>  
 machining\_activity

### 5.1.2.5.3 finishing\_or\_machining\_operations to machining\_activity (as machining\_operation)

AIM element: PATH  
Reference Path: finishing\_or\_machining\_operations<=  
property\_definition  
action\_method\_item=property\_definition  
action\_method\_item<-  
applied\_action\_method\_item.items[i]  
applied\_action\_method\_item<=  
action\_method\_assignment  
{action\_method\_assignment.role->  
action\_method\_role.name='machining operation'}  
action\_method\_assignment.assigned\_action\_method->  
action\_method=>  
process\_plan\_activity=>  
non\_casting\_operation=>  
machining\_activity

### 5.1.2.6 Heat\_treat\_requirement

AIM element: heat\_treat\_requirement  
Source: ISO 10303-223  
Reference Path: heat\_treat\_requirement<=  
customer\_casting\_requirement<=  
property\_definition

#### 5.1.2.6.1 additional\_requirements

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: heat\_treat\_requirement<=  
process\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'additional requirements'}  
representation\_item =>  
descriptive\_representation\_item

### 5.1.2.6.2 softening\_anneal\_required

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: heat\_treat\_requirement<=  
 process\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'softening anneal required'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'true')  
 (descriptive\_representation\_item.description = 'false')}

### 5.1.2.6.3 stress\_relief\_required

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: heat\_treat\_requirement<=  
 process\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'stress relief required'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'true')  
 (descriptive\_representation\_item.description = 'false')}

#### 5.1.2.6.4 heat\_treat\_requirement to organization (as where\_heat\_treat\_is\_to\_be\_performed)

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: heat\_treat\_requirement<=  
process\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
organization\_item=property\_definition  
organization\_item <-  
applied\_organization\_assignment.items [i]  
applied\_organization\_assignment <=  
organization\_assignment  
{organization\_assignment.role->  
organization\_role  
organization\_role.name='activity organization id'}  
organization\_assignment.assigned\_organization ->  
organization

#### 5.1.2.6.5 heat\_treat\_requirement to process\_property (as heat\_treatment\_per\_design\_data)

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-41  
Reference Path: heat\_treat\_requirement<=  
process\_requirement<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
{property\_definition.name = 'process property'}

**5.1.2.6.6 heat\_treat\_requirement to specification (as heat\_treat\_specification)**

AIM element: PATH  
 Reference Path: heat\_treat\_requirement<=  
 process\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 document\_reference\_item =property\_definition  
 document\_reference\_item <=  
 applied\_document\_reference.items[i]  
 applied\_document\_reference <=  
 document\_reference  
 document\_reference.assigned\_document ->  
 document  
 {document.kind->  
 document\_type  
 document\_type.product\_data\_type= 'heat treat specification'}

**5.1.2.6.7 heat\_treat\_requirement to mechanical\_property\_requirements (as mechanical\_property)**

AIM element: PATH  
 Reference Path: heat\_treat\_requirement<=  
 process\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition<=  
 property\_definition\_relationship.relying\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 mechanical\_property\_requirements

**5.1.2.7 Inspection\_or\_test\_requirement**

AIM element: inspection\_or\_test\_requirement  
 Source: ISO 10303-223  
 Reference Path: inspection\_or\_test\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition

### 5.1.2.7.1 validation\_id

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-41  
Reference Path: inspection\_or\_test\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='validation id'  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.2.7.2 inspection\_or\_test\_requirement to descriptive\_parameter (as test\_name)

AIM element: PATH  
Reference Path: inspection\_or\_test\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='test name'  
descriptive\_representation\_item



### 5.1.2.7.3 inspection\_or\_test\_requirement to numeric\_parameter(as frequency)

AIM element: PATH

Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)

Reference Path: inspection\_or\_test\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definitgion  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='frequency'  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

#### 5.1.2.7.4 inspection\_or\_test\_requirement to numeric\_parameter(as sampling\_run\_size)

AIM element: PATH

Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)

Reference Path: inspection\_or\_test\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='sampling run size'  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit  
{measure\_with\_unit.unit\_component->  
unit=named\_unit  
named\_unit=>  
context\_dependent\_unit  
context\_dependent\_unit.name='count'}  
measure\_with\_unit.value\_component ->  
measure\_value  
measure\_value = count\_measure  
count\_measure

#### 5.1.2.7.5 inspection\_or\_test\_requirement to customer\_casting\_requirement (as inspection\_against)

AIM element: PATH

Reference Path: inspection\_or\_test\_requirement<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.relying\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
customer\_casting\_requirement

### 5.1.2.7.6 inspection\_or\_test\_requirement to specification (as production\_ - quality\_requirement\_standard)

AIM element: PATH  
 Reference Path: inspection\_or\_test\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 document\_reference\_item = property\_definition  
 document\_reference\_item <=  
 applied\_document\_reference.items[i]  
 applied\_document\_reference <=  
 document\_reference  
 document\_reference.assigned\_document ->  
 document  
 {document.kind->  
 document\_type  
 document\_type.product\_data\_type= 'production quality requirement standard'}

### 5.1.2.8 Inspection\_plan

AIM element: inspection\_plan  
 Source: ISO 10303-223  
 Reference Path: inspection\_plan<=  
 product\_definition\_process<=  
 action

#### 5.1.2.8.1 correction\_description

AIM element: action.description  
 Source: ISO 10303-41  
 Reference Path: inspection\_plan<=  
 product\_definition\_process<=  
 action  
 action.description

### 5.1.2.8.2 inspection\_plan to descriptive\_parameter(as corrective\_action)

AIM element: PATH  
Reference Path: inspection\_plan<=  
product\_definition\_process<=  
action  
characterized\_action\_definition=action  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='test name'  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'weld')}  
(descriptive\_representation\_item.description = 'impregnation')}

### 5.1.2.8.3 inspection\_plan to descriptive\_parameter(as correction\_procedures)

AIM element: PATH  
Reference Path: inspection\_plan<=  
product\_definition\_process<=  
action  
characterized\_action\_definition=action  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='correction procedures'  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'dimensional')}  
{(descriptive\_representation\_item.description = 'visual')}  
(descriptive\_representation\_item.description = 'internal')}

#### 5.1.2.8.4 inspection\_plan to inspection\_or\_test\_requirement (as customer\_ - requirements)

AIM element: PATH  
 Reference Path: inspection\_plan<=  
                   product\_definition\_process<=  
                   action  
                   characterized\_action\_definition=action  
                   characterized\_action\_definition<-  
                   action\_property.definition  
                   action\_property=>  
                   customer\_casting\_requirement=>  
                   inspection\_or\_test\_requirement

#### 5.1.2.8.5 inspection\_plan to special\_inspection\_requirement (as customer\_ - requirements)

AIM element: PATH  
 Reference Path: inspection\_plan<=  
                   product\_definition\_process<=  
                   action  
                   characterized\_action\_definition=action  
                   characterized\_action\_definition<-  
                   action\_property.definition  
                   action\_property=>  
                   customer\_casting\_requirement=>  
                   special\_inspection\_requirement

#### 5.1.2.9 Machining\_allowance

AIM element: machine\_allowance  
 Source: ISO 10303-223  
 Reference Path: machine\_allowance<=  
                   property\_definition

### 5.1.2.9.1 machining\_allowance to manufacturing\_feature (as associated\_feature)

AIM element: PATH  
Reference Path: machine\_allowance<=  
property\_definition<-  
property\_definition\_relationship.relatiing\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition  
property\_definition.definition->  
instanced\_feature<=  
feature\_definiton

### 5.1.2.9.2 machining\_allowance to numeric\_parameter (as allowance\_value)

AIM element: PATH  
Reference Path: machine\_allowance<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.representation->  
representation.items[i] ->  
representation\_item  
representation\_item.name='allowance value'  
(measure\_representation\_item)

### 5.1.2.10 Mechanical\_property\_requirements

AIM element: mechanical\_property\_requirements  
Source: ISO 10303-223  
Reference Path: mechanical\_property\_requirements<=  
property\_definition

### 5.1.2.10.1 chemical\_analysis\_certification

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: mechanical\_property\_requirements<=  
 process\_requirement<=  
 customer\_casting\_requirement<=  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'chemical analysis certification'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'true')  
 (descriptive\_representation\_item.description = 'false')}

### 5.1.2.10.2 mechanical\_properties\_certification

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: mechanical\_property\_requirements<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'mechanical properties certification'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'true')  
 (descriptive\_representation\_item.description = 'false')}

### 5.1.2.10.3 tensile\_strength

AIM element: PATH  
Reference Path: mechanical\_property\_requirements<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='tensile strength'  
descriptive\_representation\_item

### 5.1.2.10.4 yeild\_strength

AIM element: PATH  
Reference Path: mechanical\_property\_requirements<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='yeild strength'  
descriptive\_representation\_item

### 5.1.2.10.5 mechanical\_property\_requirements to material\_hardness (as hardness)

AIM element: PATH  
Reference Path: mechanical\_property\_requirements<=  
property\_definition<-  
property\_definition\_relationship.relying\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
material\_property



### 5.1.2.10.6 mechanical\_property\_requirements to specification (as requirement\_specification)

AIM element: PATH  
 Reference Path: mechanical\_property\_requirements<=  
                   property\_definition  
                   document\_reference\_item = property\_definition  
                   document\_reference\_item <-  
                   applied\_document\_reference.items[i]  
                   applied\_document\_reference <=  
                   document\_reference  
                   document\_reference.assigned\_document ->  
                   document  
                   {document.kind->  
                   document\_type  
                   document\_type.product\_data\_type= 'requirement specification'}

### 5.1.2.11 Mechanical\_stress

AIM element: mechanical\_stress  
 Source: ISO 10303-223  
 Reference Path: mechanical\_stress<=  
                   property\_definition

#### 5.1.2.11.1 mechanical\_stress to descriptive\_parameter(as type\_of\_stress)

AIM element: PATH  
 Reference Path: mechanical\_stress<=  
                   property\_definition  
                   represented\_definition=property\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation.used\_representation->  
                   representation  
                   representation.items[i] ->  
                   representation\_item  
                   representation\_item.name=' type of stress'  
                   descriptive\_representation\_item

### 5.1.2.11.2 mechanical\_stress to numeric\_parameter(as maximum\_psi\_design\_stress)

AIM element: PATH  
Reference Path: mechanical\_stress<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='maximum psi design stress'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.2.11.3 mechanical\_stress to numeric\_parameter(as percent\_safety\_factor\_required)

AIM element: PATH  
Reference Path: mechanical\_stress<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='percent safety factor required'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.2.12 Mismatch\_tolerance

AIM element: mismatch\_tolerance  
 Source: ISO 10303-223  
 Reference Path: mismatch\_tolerance<=  
 property\_definition

#### 5.1.2.12.1 mismatch\_tolerance to numeric\_parameter(as maximum\_amount)

AIM element: PATH  
 Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: mismatch\_tolerance<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 (representation.description = 'maximum amount')  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

#### 5.1.2.12.2 mismatch\_tolerance to parting\_surface(as with\_respect\_to)

AIM element: PATH  
 Reference Path: mismatch\_tolerance<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation =>  
 shape\_representation=>  
 parting\_surface

### 5.1.2.13 Pattern\_equipment\_available

AIM element: pattern\_equipment\_available  
Source: ISO 10303-223  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition

#### 5.1.2.13.1 core\_box\_cavity\_id

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
(representation\_item.name = 'core box cavity id')  
representation\_item =>  
descriptive\_representation\_item

#### 5.1.2.13.2 core\_box\_id

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
(representation\_item.name = 'core box id')  
representation\_item =>  
descriptive\_representation\_item

### 5.1.2.13.3 number\_of\_core\_boxes

AIM element: PATH  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item =>  
 (representation\_item.name = 'number of core boxes')  
 (measure\_representation\_item)

### 5.1.2.13.4 number\_of\_cores\_per\_box

AIM element: PATH  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item =>  
 (representation\_item.name = 'number of cores per box')  
 measure\_representation\_item

### 5.1.2.13.5 number\_of\_cores\_per\_casting

AIM element: PATH  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item =>  
(representation\_item.name = 'number of cores per casting')  
measure\_representation\_item

### 5.1.2.13.6 number\_of\_patterns

AIM element: PATH  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item =>  
(representation\_item.name = 'number of patterns')  
measure\_representation\_item

**5.1.2.13.7 condition**

AIM element:       descriptive\_representation\_item.description  
Source:             ISO 10303-45  
Reference Path:     pattern\_equipment\_available<=  
                      customer\_casting\_requirement<=  
                      property\_definition  
                      represented\_definition=property\_definition  
                      represented\_definition<-  
                      property\_definition\_representation.definition  
                      property\_definition\_representation.used\_representation->  
                      representation  
                      representation.items[i] ->  
                      representation\_item  
                      representation\_item =>  
                      (representation\_item.name = 'condition')  
                      descriptive\_representation\_item  
                      (descriptive\_representation\_item.description = 'good')  
                      (descriptive\_representation\_item.description = 'bad')  
                      (descriptive\_representation\_item.description = 'fair')  
                      (descriptive\_representation\_item.description = 'new')}

### 5.1.2.13.8 core\_box\_material

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.related\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.name= 'core box material'  
property\_definition\_relationship.relating\_property\_definition->  
property\_definition  
characterized\_product\_definition = property\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation  
{(material\_designation.name = 'plaster')  
(material\_designation.name = 'expandable polystyrene')  
(material\_designation.name = 'iron')  
(material\_designation.name = 'soft wood')  
(material\_designation.name = 'shell')  
(material\_designation.name = 'aluminum')  
(material\_designation.name = 'epoxy')  
(material\_designation.name = 'wood metal reinforced')  
(material\_designation.name = 'brass')  
(material\_designation.name = 'hard wood')  
(material\_designation.name = 'urethane')  
(material\_designation.name = 'other')  
(material\_designation.name = 'steel')  
(material\_designation.name = 'ceramic')  
(material\_designation.name = 'polymethylmethacrylate')  
(material\_designation.name = 'plywood')} }



### 5.1.2.13.9 pattern\_impression\_id

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item=>  
 (representation\_item.name = 'pattern impression id')  
 descriptive\_representation\_item

### 5.1.2.13.10 pattern\_material

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.related\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.name= 'core box material'  
property\_definition\_relationship.relating\_property\_definition->  
property\_definition  
characterized\_product\_definition = property\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation  
{(material\_designation.name = 'plaster')  
(material\_designation.name = 'expandable polystyrene')  
(material\_designation.name = 'iron')  
(material\_designation.name = 'soft wood')  
(material\_designation.name = 'shell')  
(material\_designation.name = 'aluminum')  
(material\_designation.name = 'epoxy')  
(material\_designation.name = 'wood metal reinforced')  
(material\_designation.name = 'brass')  
(material\_designation.name = 'hard wood')  
(material\_designation.name = 'urethane')  
(material\_designation.name = 'other')  
(material\_designation.name = 'steel')  
(material\_designation.name = 'ceramic')  
(material\_designation.name = 'polymethylmethacrylate')  
(material\_designation.name = 'plywood')} }

### 5.1.2.13.11 core\_per\_core\_box\_per\_assembly

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 (representation\_item.name = 'core per core box per assembly')  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'same core')  
 (descriptive\_representation\_item.description = 'different core')}

### 5.1.2.13.12 pattern

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 (representation\_item.name = 'pattern')  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'loose pattern with follow on board')  
 (descriptive\_representation\_item.description = 'loose solid pattern')  
 (descriptive\_representation\_item.description = 'loose split pattern')  
 (descriptive\_representation\_item.description = 'mounted cope and drag boards')  
 (descriptive\_representation\_item.description = 'mounted cope and drag flaskless')  
 (descriptive\_representation\_item.description = 'mounted cope and drag inserts')  
 (descriptive\_representation\_item.description = 'mounted snapboard')  
 (descriptive\_representation\_item.description = 'mounted match plate')  
 (descriptive\_representation\_item.description = 'sweep pattern')}

### 5.1.2.13.13 **pattern\_equipment\_available to flask (as flask\_size\_for\_ - mounted\_patterns)**

AIM element: PATH

Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.related\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.name= 'flask size for mounted patterns'  
property\_definition\_relationship.relying\_property\_definition->  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
sand\_cast\_feature\_definition=>  
flask

### 5.1.2.13.14 **pattern\_equipment\_available to flask (as flask\_size\_for\_plate\_ - patterns)**

AIM element: PATH

Reference Path: pattern\_equipment\_available<=  
customer\_casting\_requirement<=  
property\_definition<-  
property\_definition\_relationship.related\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.name= 'flask size for plate patterns'  
property\_definition\_relationship.relying\_property\_definition->  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
sand\_cast\_feature\_definition=>  
flask

**5.1.2.13.15 pattern\_equipment\_available to mould\_box (as flaskless\_pattern)**

AIM element: PATH  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition<-  
 property\_definition\_relationship.related\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.name= 'flaskless pattern'  
 property\_definition\_relationship.relying\_property\_definition->  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_object  
 characterized\_object=>  
 sand\_cast\_feature\_definition=>  
 mould\_box

**5.1.2.13.16 pattern\_equipment\_available to in\_facility\_location (as tooling\_location)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: pattern\_equipment\_available<=  
 customer\_casting\_requirement<=  
 property\_definition<-<-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 in\_facility\_location

### 5.1.2.13.17 **pattern\_equipment\_available to person\_in\_organization (as tooling\_location)**

AIM element: PATH  
Reference Path: production\_core\_box<=  
sand\_cast\_design\_feature<=  
casting\_instanced\_feature<=  
shape\_aspect<-  
applied\_person\_and\_organization\_assignment.items  
applied\_person\_and\_organization\_assignment<=  
person\_and\_organization\_assignment  
{person\_and\_organization\_assignment.role->  
person\_and\_organization\_role  
person\_and\_organization\_role.name='core box stored'}  
person\_and\_organization\_assignment.assigned\_person\_and\_organization->  
person\_and\_organization

### 5.1.2.14 **Payment\_and\_shipping**

AIM element: payment\_and\_shipping  
Source: ISO 10303-223  
Reference Path: payment\_and\_shipping <=  
customer\_casting\_requirement<=  
property\_definition

#### 5.1.2.14.1 **customer\_payment**

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: payment\_and\_shipping<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'customer payment'}  
representation\_item =>  
descriptive\_representation\_item

### 5.1.2.14.2 packaging\_or\_creating

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: payment\_and\_shipping<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definitgion  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'packaging or creating'}  
 representation\_item =>  
 descriptive\_representation\_item

### 5.1.2.14.3 ship\_casting

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: payment\_and\_shipping<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definitgion  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'ship casting'}  
 representation\_item =>  
 descriptive\_representation\_item

### 5.1.2.15 Process\_requirement

AIM element: process\_requirement  
 Source: ISO 10303-41  
 Reference Path: process\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition

### 5.1.2.15.1 process\_requirement to descriptive\_parameter(as process\_name)

AIM element: PATH  
Reference Path: process\_requirement<=  
customer\_casting\_requirement =  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<=  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
(representation.description = 'process name')  
(descriptive\_representation\_item)

### 5.1.2.16 Reporting\_requirement

AIM element: reporting\_requirement  
Source: ISO 10303-223  
Reference Path: reporting\_requirement<=  
customer\_casting\_requirement<=  
property\_definition

#### 5.1.2.16.1 reporting\_requirement to date (as date\_to\_be\_reported)

AIM element: date  
Source: ISO 10303-41  
Rules dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: reporting\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
date\_item = property\_definition  
date\_item <=  
applied\_date\_assignment.items[i]  
applied\_date\_assignment <=  
{date\_assignment  
date\_assignment.role ->  
date\_role  
date\_role.name = 'required delivery date'}  
date\_assignment  
date\_assignment.assigned\_date ->  
date



### 5.1.2.16.2 reporting\_requirement to customer\_casting\_requirement (as based\_on)

AIM element: PATH  
 Reference Path: reporting\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition <-  
 property\_definition\_relationship.related\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.relatng\_property\_definition ->  
 property\_definition=>  
 customer\_casting\_requirement

### 5.1.2.17 Special\_inspection\_requirement

AIM element: special\_inspection\_requirement  
 Source: ISO 10303-223  
 Reference Path: special\_inspection\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition

#### 5.1.2.17.1 inspection\_description

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: special\_inspection\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definitgion  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'inspection description'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'ultrasonic')  
 {(descriptive\_representation\_item.description = 'other testing method')  
 {(descriptive\_representation\_item.description = 'pressure test')  
 {(descriptive\_representation\_item.description = 'x ray radiography')  
 {(descriptive\_representation\_item.description = 'welding')  
 {(descriptive\_representation\_item.description = 'liquid penetrant')  
 (descriptive\_representation\_item.description = 'magnetic particle')}}

### 5.1.2.17.2 other\_testing\_method

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: special\_inspection\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'other testing method'}  
representation\_item =>  
descriptive\_representation\_item

### 5.1.2.17.3 performed\_by\_customer

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: special\_inspection\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'performed by customer'}  
representation\_item =>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'true')  
(descriptive\_representation\_item.description = 'false')}

### 5.1.2.17.4 special\_inspection\_requirement to specification (as inspection\_ - requirements)

AIM element: PATH  
 Reference Path: special\_inspection\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 document\_reference\_item = property\_definition  
 document\_reference\_item <-  
 applied\_document\_reference.items[i]  
 applied\_document\_reference <=  
 document\_reference  
 document\_reference.assigned\_document ->  
 document  
 {document.kind->  
 document\_type  
 document\_type.product\_data\_type= 'inspection specification'}

### 5.1.2.18 Tolerance\_requirement

AIM element: tolerance\_requirement  
 Source: ISO 10303-223  
 Reference Path: tolerance\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition

#### 5.1.2.18.1 dimension\_method

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: tolerance\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 represented\_definition=property\_definitgion  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'dimension method'}  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'commercial practice')  
 {(descriptive\_representation\_item.description = 'specification data')  
 (descriptive\_representation\_item.description = 'drawing data')}

### 5.1.2.18.2 gaging\_fixtures\_available

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: tolerance\_requirement<=  
process\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'gaging fixtures available'}  
representation\_item =>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'true')  
(descriptive\_representation\_item.description = 'false')}

### 5.1.2.18.3 tolerance\_requirement to dimensional\_tolerance(as tolerance\_-based\_on)

AIM element: PATH  
Reference Path: tolerance\_requirement<=  
customer\_casting\_requirement<=  
property\_definition  
represented\_definition=property\_definitgion  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'tolerance based on'}  
representation\_item =>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'mismatch')  
(descriptive\_representation\_item.description = 'dimensional')  
(descriptive\_representation\_item.description = 'geometric')}

#### 5.1.2.18.4 tolerance\_requirement to checking\_aid\_tool (as gaging\_requirement)

AIM element: PATH  
 Reference Path: tolerance\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition <-  
 property\_definition\_relationship.related\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.relating\_property\_definition ->  
 property\_definition=>  
 checking\_aid\_tool

#### 5.1.2.18.5 tolerance\_requirement to mismatch\_tolerance ( as mismatch)

AIM element: PATH  
 Reference Path: reporting\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition <-  
 property\_definition\_relationship.related\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.relating\_property\_definition ->  
 property\_definition=>  
 mismatch\_tolerance

#### 5.1.2.18.6 tolerance\_requirement to specification (as dimension\_specification)

AIM element: PATH  
 Reference Path: tolerance\_requirement<=  
 customer\_casting\_requirement<=  
 property\_definition  
 document\_reference\_item = property\_definition  
 document\_reference\_item <-  
 applied\_document\_reference.items[i]  
 applied\_document\_reference <=  
 document\_reference  
 document\_reference.assigned\_document ->  
 document  
 {document.kind->  
 document\_type  
 document\_type.product\_data\_type= 'dimension specification'}

### 5.1.3 Design\_exception UoF

#### 5.1.3.1 Design\_exception\_notice

AIM element: versioned\_action\_request  
Source: ISO 10303-41

##### 5.1.3.1.1 notice\_description

AIM element: versioned\_action\_request.purpose  
Source: ISO 10303-41

##### 5.1.3.1.2 notice\_number

AIM element: versioned\_action\_request.id  
Source: ISO 10303-41

##### 5.1.3.1.3 technical\_recommendation

AIM element: action\_method  
Source: ISO 10303-41  
Reference Path: versioned\_action\_request <-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method ->  
action\_method

##### 5.1.3.1.4 design\_exception\_notice to date (issuing\_date)

AIM element: date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: versioned\_action\_request  
date\_item = versioned\_action\_request  
date\_item <-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment <=  
{date\_assignment  
date\_assignment.role ->  
date\_role  
date\_role.name = 'issuing date'}  
date\_assignment  
date\_assignment.assigned\_date ->  
date

### 5.1.3.1.5 design\_exception\_notice to engineering\_change\_proposal (as issues)

AIM element: PATH  
 Reference Path: versioned\_action\_request <-  
 action\_request\_solution.request  
 action\_request\_solution  
 action\_request\_solution.method ->  
 action\_method  
 document\_reference\_item = action\_method  
 document\_reference\_item <-  
 applied\_document\_reference.items[i]  
 applied\_document\_reference <=  
 document\_reference  
 document\_reference.assigned\_document ->  
 document  
 {document  
 document.kind ->  
 document\_type  
 document\_type.product\_data\_type = 'engineering change proposal'}

### 5.1.3.1.6 design\_exception\_notice to part\_version (as discrepant\_part)

AIM element: PATH  
 Reference Path: versioned\_action\_request <-  
 action\_request\_assignment.assigned\_action\_request  
 action\_request\_assignment =>  
 applied\_action\_request\_assignment  
 applied\_action\_request\_assignment.items[i] ->  
 action\_request\_item  
 action\_request\_item = product\_definition\_formation  
 product\_definition\_formation

## 5.1.3.2 Engineering\_change\_order

AIM element: action\_directive  
 Source: ISO 10303-41

### 5.1.3.2.1 change\_order\_number

AIM element: action\_directive.name  
 Source: ISO 10303-41

### 5.1.3.2.2 engineering\_change\_order to part\_version (as new\_version)

AIM element: PATH  
Reference Path: action\_directive <-  
directed\_action.directive  
directed\_action <=  
executed\_action <=  
action <-  
action\_assignment.assigned\_action  
action\_assignment =>  
applied\_action\_assignment  
applied\_action\_assignment.items[i] ->  
action\_item  
action\_item = product\_definition\_formation  
product\_definition\_formation

### 5.1.3.3 Engineering\_change\_proposa

AIM element: document  
Source: ISO 10303-41  
Reference Path: {document  
document.kind ->  
document\_type  
document\_type.product\_data\_type = 'engineering change proposal'

#### 5.1.3.3.1 change\_proposal\_number

AIM element: document.id  
Source: ISO 10303-41



### 5.1.3.3.2 engineering\_change\_proposal to engineering\_change\_order (as incorporated\_proposal)

AIM element: PATH  
 Reference Path: document<-  
                   document\_reference.assigned\_document  
                   document\_reference=>  
                   applied\_document\_reference  
                   applied\_document\_reference.items[i]->  
                   document\_reference\_item  
                   document\_reference\_item = action\_method  
                   action\_method<-  
                   action.chosen\_method  
                   action=>  
                   executed\_action=>  
                   directed\_action  
                   directed\_action.directive->  
                   action\_directive

### 5.1.3.4 Request\_for\_clarification

AIM element: versioned\_action\_request  
 Source: ISO 10303-41  
 Reference Path: {versioned\_action\_request  
                   versioned\_action\_request.purpose='request for clarification'}

#### 5.1.3.4.1 description

AIM element: versioned\_action\_request.description  
 Source: ISO 10303-41  
 Reference Path: versioned\_action\_request  
                   versioned\_action\_request.description

### 5.1.3.4.2 request\_for\_clarification\_to\_date (as request\_date)

AIM element: date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: versioned\_action\_request  
date\_item = versioned\_action\_request  
date\_item <-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment <=  
{date\_assignment  
date\_assignment.role ->  
date\_role  
date\_role.name = 'request date'}  
date\_assignment  
date\_assignment.assigned\_date ->  
date

### 5.1.3.4.3 request\_for\_clarification\_to\_person\_in\_organization (as requested\_by)

AIM element: PATH  
Reference Path: versioned\_action\_request  
person\_and\_organization\_item = versioned\_action\_request  
person\_and\_organization\_item <-  
applied\_person\_and\_organization\_assignment.items[i]  
applied\_person\_and\_organization\_assignment <=  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.assigned\_person\_and\_organization ->  
person\_and\_organization

### 5.1.3.4.4 request\_for\_clarification\_to\_part\_version (as requested\_part)

AIM element: PATH  
Reference Path: versioned\_action\_request <-  
action\_request\_assignment.assigned\_action\_request  
action\_request\_assignment =>  
applied\_action\_request\_assignment  
applied\_action\_request\_assignment.items[i] ->  
action\_request\_item  
action\_request\_item = product\_definition\_formation  
product\_definition\_formation

## 5.1.4 Die\_mould features UoF

### 5.1.4.1 Cooling\_port

AIM element: cooling\_port  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: cooling\_port<=  
die\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.4.1.1 male\_or\_female

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: cooling\_port<=  
die\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'male or female'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description  
{(descriptive\_representation\_item.description = 'male')  
(descriptive\_representation\_item.description = 'female')}

### 5.1.4.1.2 cooling\_port to item\_size (as cooling\_size)

```

AIM element:      PATH
Reference Path:  cooling_port<=
                 die_design_feature
                 casting_feature_definition<=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'cooling port occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 feature_component_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description = 'cooling port size usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relying_shape_aspect ->
                 shape_aspect =>
                 casting_feature_size

```

### 5.1.4.1.3 cooling\_port to planar\_element(as port\_placement)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: cooling\_port<=  
die\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
{representation  
representation.name = 'port location'}  
representation =>  
shape\_representation=>  
planar\_shape\_representation

### 5.1.4.2 Die\_design\_feature

AIM element: die\_design\_feature  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: die\_design\_feature<=  
casting\_feature\_definition  
[casting\_feature\_definition=>  
casting\_instanced\_feature]

### 5.1.4.2.1 die\_design\_feature to orientation (as placement)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: die\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'orientation'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.4.3 Die\_cast\_master

AIM element: die\_cast\_master  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: die\_cast\_master<=  
 die\_cast\_tooling<=  
 casting\_production\_definition  
 product\_definition  
 product\_definition.frame\_of\_reference->  
 product\_definition\_context<=  
 application\_context\_element  
 application\_context\_element.name='functional definition'

### 5.1.4.3.1 master\_shrink\_factor

AIM element: length\_measure\_with\_unit  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: die\_cast\_master<=  
die\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition<=  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
core\_shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'master shrink factor'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}



### 5.1.4.3.2 die\_cast\_master to production\_die\_cast\_mould (as female\_ - component)

AIM element: PATH

Reference Path: die\_cast\_master<=  
 die\_cast\_tooling<=  
 casting\_production\_definition  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'die cast tooling occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description\  
 ='production die cast mould female reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 production\_tool=>  
 die\_design\_feature=>  
 production\_die\_cast\_mould

### 5.1.4.3.3 die\_cast\_master to production\_die\_mould (as male\_component)

AIM element: PATH  
Reference Path: die\_cast\_master<=  
die\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'die cast tooling occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description\  
='production die cast mould male reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
production\_tool=>  
die\_design\_feature=>  
production\_die\_cast\_mould

#### 5.1.4.3.4 die\_cast\_master to machining\_allowance (as master\_allowance\_on\_master\_tooling)

AIM element: PATH  
 Reference Path: die\_cast\_master<=  
                   die\_cast\_tooling<=  
                   casting\_production\_definition  
                   product\_definition  
                   characterized\_product\_definition = product\_definition  
                   characterized\_product\_definition  
                   characterized\_definition = characterized\_product\_definition  
                   characterized\_definition <-  
                   property\_definition.definition  
                   property\_definition=>  
                   machining\_allowance

#### 5.1.4.4 Die\_cast\_tooling

AIM element: die\_cast\_tooling  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
        subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: die\_cast\_tooling<=  
                   casting\_production\_definition  
                   product\_definition  
                   product\_definition.frame\_of\_reference->  
                   product\_definition\_context<=  
                   application\_context\_element  
                   application\_context\_element.name='functional definition'

##### 5.1.4.4.1 tooling\_identifer

AIM element: product\_definition.description  
 Source: ISO 10303-45  
 Reference Path: die\_cast\_tooling<=  
                   casting\_production\_definition<=  
                   product\_definition  
                   product\_definition.description

### 5.1.4.4.2 die\_cast\_tooling to production\_die\_cast\_mould (as die\_equipment)

AIM element: PATH  
Reference Path: die\_cast\_tooling<=  
casting\_production\_definition<=  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'die cast tooling occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n'production die cast mould reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
production\_tool=>  
die\_design\_feature=>  
production\_die\_cast\_mould

### 5.1.4.4.3 die\_cast\_tooling to material (as material)

AIM element: material\_designation.name  
 Source: ISO 10303-45  
 Reference Path: die\_cast\_tooling<=  
   casting\_production\_definition  
   product\_definition  
   product\_definition\_relationship.relating\_product\_definition  
   {product\_definition\_relationship.description='raw material'  
   product\_definition\_relationship =>  
   product\_definition\_usage =>  
   make\_from\_usage\_option}  
   product\_definition\_relationship  
   product\_definition\_relationship.related\_product\_definition ->  
   product\_definition  
   {product\_definition  
   characterized\_product\_definition = product\_definition  
   characterized\_product\_definition  
   characterized\_definition = characterized\_product\_definition  
   characterized\_definition <-  
   material\_designation.definitions[i]  
   material\_designation}

### 5.1.4.4.4 die\_cast\_tooling to in\_facility\_location (as tool\_location)

AIM element: PATH  
 Reference Path: die\_cast\_tooling<=  
   casting\_production\_definition  
   product\_definition  
   characterized\_product\_definition=product\_definition  
   characterized\_product\_definition  
   characterized\_definition = characterized\_product\_definition  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   representation =>  
   in\_facility\_location

#### 5.1.4.4.5 die\_cast\_tooling to person\_and\_organization (as tool\_location)

AIM element: PATH  
Reference Path: die\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
person\_and\_organization\_item = product\_definition  
person\_and\_organization\_item<=  
applied\_person\_and\_organization\_assignment.items[i]  
applied\_person\_and\_organization\_assignment<=  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.assignment\_person\_and\_organization->  
person\_and\_organization

#### 5.1.4.5 Die\_clamping

AIM element: die\_clamping  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: die\_clamping<=  
die\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.4.5.1 die\_clamping to orientation (as die\_clamping\_positions)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: die\_clamping <=  
 die\_design\_feature <=  
 casting\_feature\_definition =>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 {property\_definition =>  
 product\_definition\_shape}  
 property\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'clamping position'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.4.6 Die\_core

AIM element: die\_core  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: die\_core <=  
 die\_design\_feature  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.4.6.1 core\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: die\_core<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition <-  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation =>  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'core type'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {(descriptive\_representation\_item.description = 'fixed core')  
 (descriptive\_representation\_item.description = 'hydraulic slide')  
 (descriptive\_representation\_item.description = 'mechanical slide')  
 (descriptive\_representation\_item.description = 'sand core')  
 (descriptive\_representation\_item.description = 'salt core')  
 (descriptive\_representation\_item.description = 'ceramic')}  
 }



### 5.1.4.6.2 metal\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: die\_core<=  
   die\_design\_feature  
   casting\_feature\_definition =>  
   casting\_instanced\_feature}  
   feature\_definition <=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   material\_designation.definitions[i]  
   material\_designation  
   material\_designation.name

### 5.1.4.6.3 die\_core to drafted\_surface (as draft)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: die\_core<=  
   die\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition <-  
   {property\_definition=>  
   product\_definition\_shape}  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   {representation =>  
   shape\_representation =>  
   shape\_representation\_with\_parameters}  
   representation =>  
   shape\_representation=>  
   drafted\_surface

#### 5.1.4.6.4 die\_core to item\_size (as guide\_pin)

AIM element: PATH  
Reference Path: die\_core<=  
die\_design\_feature  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'die core occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'guide pin size usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

### 5.1.4.6.5 die\_core to parting\_surface (as parting)

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: die\_core<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
shape\_representation=>  
parting\_surface

#### 5.1.4.6 die\_core to shape\_aspect (as outside\_shape\_of\_die\_mould)

AIM element: PATH  
Source: ISO 10303-223  
Reference Path: die\_core<=  
sand\_cast\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'die core occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='die core area shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect

#### 5.1.4.7 Die\_master

AIM element: die\_master  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: die\_master<=  
die\_mould\_tooling<=  
casting\_production\_definition  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

### 5.1.4.7.1 master\_shrink\_factor

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: die\_master<=  
 die\_mould\_tooling<=  
 casting\_production\_definition  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition<-  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 core\_shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'master shrink factor'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.4.7.2 die\_master to production\_die\_mould (as female\_component)

AIM element: PATH

```
Reference Path: die_master<=  
casting_production_definition  
product_definition  
characterized_product_definition = product_definition  
characterized_product_definition  
characterized_definition = characterized_product_definition  
characterized_definition <-  
property_definition.definition  
property_definition =>  
product_definition_shape <-  
shape_aspect.of_shape  
{shape_aspect  
shape_aspect.description = 'die master occurrence'}  
shape_aspect <-  
shape_aspect_relationship.related_shape_aspect  
{[shape_aspect_relationship =>  
shape_defining_relationship]  
[shape_aspect_relationship  
shape_aspect_relationship.description=\  
'production die mould female reference usage']}  
shape_aspect_relationship  
shape_aspect_relationship.relate_shape_aspect ->  
shape_aspect =>  
casting_instanced_feature=>  
casting_feature_definition=>  
die_design_feature=>  
production_tool=>  
production_die_mould}
```

### 5.1.4.7.3 die\_master to production\_die\_mould (as male\_component)

AIM element: PATH  
 Reference Path: die\_master<=  
 casting\_production\_definition  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'die master occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description =\  
 'production die mould male reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 die\_design\_feature=>  
 production\_tool=>  
 production\_die\_mould}

### 5.1.4.7.4 die\_master to machining\_allowance (as master\_allowance\_on\_machinable\_surfaces)

AIM element: PATH  
 Reference Path: die\_master<=  
 casting\_production\_definition  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition=>  
 machining\_allowance

### 5.1.4.8 Die\_mould\_tooling

AIM element: die\_mould\_tooling  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: die\_mould\_tooling<=  
casting\_production\_definition<=  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

#### 5.1.4.8.1 tooling\_identifier

AIM element: product\_definition.description  
Source: ISO 10303-45  
Reference Path: die\_mould\_tooling<=  
casting\_production\_definition<=  
product\_definition  
product\_definition.description

#### 5.1.4.8.2 die\_mould\_tooling to material (as material\_definition)

AIM element: PATH  
Reference Path: die\_mould\_tooling<=  
casting\_production\_definition  
product\_definition  
product\_definition\_relationship.relying\_product\_definition  
{product\_definition\_relationship.description='raw material'  
product\_definition\_relationship =>  
product\_definition\_usage =>  
make\_from\_usage\_option}  
product\_definition\_relationship  
product\_definition\_relationship.related\_product\_definition ->  
product\_definition  
{product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation}  
material\_designation.name



### 5.1.4.8.3 die\_mould\_tooling to production\_die\_mould (as die\_equipment)

AIM element: PATH  
 Reference Path: die\_mould\_tooling<=  
 casting\_production\_definition<=  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'die mould tooling occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description=\n  
 'production die mould reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 production\_tool=>  
 die\_design\_feature=>  
 production\_die\_mould

#### 5.1.4.8.4 die\_mould\_tooling to in\_facility\_location (as tooling\_location)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: die\_mould\_tooling<=  
casting\_production\_definition  
product\_definition  
characterized\_product\_definition=product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition <-  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
in\_facility\_location

#### 5.1.4.8.5 die\_mould\_tooling to person\_and\_organization (as tooling\_location)

AIM element: PATH  
Reference Path: die\_mould\_tooling<=  
casting\_production\_definition  
product\_definition  
person\_and\_organization\_item = product\_definition  
person\_and\_organization\_item<-  
applied\_person\_and\_organization\_assignment.items[i]  
applied\_person\_and\_organization\_assignment<=  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.assigment\_person\_and\_organization->  
person\_and\_organization

### 5.1.4.9 Die\_mould\_vent

AIM element: die\_mould\_vent  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: die\_mould\_vent<=  
 die\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.4.9.1 die\_mould\_vent to ejector\_pin (as ejector\_pin\_definition)

AIM element: PATH  
 Reference Path: die\_mould\_vent<=  
 die\_design\_feature  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'die mould vent occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='ejector pin reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relating\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 ejector\_design\_feature=>  
 ejector\_pin

### 5.1.4.10 Production\_die\_cast\_mould

AIM element: production\_die\_cast\_mould  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
          subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: production\_die\_cast\_mould<=  
                  die\_design\_feature<=  
                  production\_tool<=  
                  casting\_design\_feature<=  
                  {casting\_feature\_definition =>  
                  casting\_instanced\_feature}  
                  casting\_feature\_definition <=  
                  characterized\_object

### 5.1.4.10.1 cooling\_requirement

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_die\_cast\_mould<=  
                   production\_tool<=  
                   die\_design\_feature<=  
                   casting\_feature\_definition=>  
                   casting\_instanced\_feature <=  
                   shape\_aspect  
                   shape\_definition = shape\_aspect  
                   shape\_definition  
                   characterized\_definition = shape\_definition  
                   characterized\_definition <-  
                   property\_definition.definition  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition <-  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   representation =>  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'cooling requirement'}  
                   representation\_item =>  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description  
                   {(descriptive\_representation\_item.description = 'TRUE')  
                   (descriptive\_representation\_item.description = 'FALSE')}

### 5.1.4.10.2 insert\_material\_type

AIM element: material\_designation  
Source: ISO 10303-45  
Reference Path: production\_die\_cast\_mould<=  
die\_design\_feature<=  
production\_tool<=  
casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation  
material\_designation.name

### 5.1.4.10.3 inserts

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_die\_cast\_mould<=  
                   production\_tool<=  
                   die\_design\_feature<=  
                   casting\_feature\_definition=>  
                   casting\_instanced\_feature <=  
                   shape\_aspect  
                   shape\_definition = shape\_aspect  
                   shape\_definition  
                   characterized\_definition = shape\_definition  
                   characterized\_definition <-  
                   property\_definition.definition  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition <-  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   representation =>  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'insert'}  
                   representation\_item =>  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description  
                   {(descriptive\_representation\_item.description = 'unit')  
                   (descriptive\_representation\_item.description = 'combination')  
                   (descriptive\_representation\_item.description = 'cast in')  
                   (descriptive\_representation\_item.description = 'none')}

#### 5.1.4.10.4 port\_connection\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'port connection type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description



### 5.1.4.10.5 die\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_die\_cast\_mould<=  
                   production\_tool<=  
                   die\_design\_feature<=  
                   casting\_feature\_definition=>  
                   casting\_instanced\_feature <=  
                   shape\_aspect  
                   shape\_definition = shape\_aspect  
                   shape\_definition  
                   characterized\_definition = shape\_definition  
                   characterized\_definition <-  
                   property\_definition.definition  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition <-  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   representation =>  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'die type'}  
                   representation\_item =>  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description  
                   {(descriptive\_representation\_item.description = 'vacuum chamber')  
                   (descriptive\_representation\_item.description = 'hot chamber')  
                   (descriptive\_representation\_item.description = 'cold chamber')}

### 5.1.4.10.6 production\_die\_cast\_mould to cooling\_port (as cooling\_type)

AIM element: PATH  
Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'die mould vent occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='ejector pin reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
cooling\_port

### 5.1.4.10.7 production\_die\_cast\_mould to item\_size (as external\_box\_dimensions)

AIM element: PATH

Reference Path: production\_die\_cast\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production die cast mould occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'production die cast mould dimension  
 usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.4.10.8 production\_die\_cast\_mould to runner (as gate\_runner)

AIM element: PATH  
Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die cast mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='gate runner reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
runner

### 5.1.4.10.9 production\_die\_cast\_mould to ingate (as gate\_runner)

AIM element: PATH  
 Reference Path: production\_die\_cast\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production die cast mould occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='gate runner reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 ingate

#### 5.1.4.10.10 production\_die\_cast\_mould to secondary\_tooling (as secondary\_tooling\_requirements)

AIM element: PATH

Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die cast mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='secondary tooling reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
secondary\_tooling

**5.1.4.10.11 production\_die\_cast\_mould to sprue (as pouring\_sprue)**

AIM element: PATH  
Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die cast mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='sprue reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
sprue

### 5.1.4.10.12 production\_die\_cast\_mould to slide (as movable\_die\_items)

AIM element: PATH  
Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die cast mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='slide reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
slide



### 5.1.4.10.13 production\_die\_cast\_mould to die\_clamping (as movable\_die\_items)

AIM element: PATH

Reference Path: production\_die\_cast\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production die cast mould occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='movable die clamping reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 die\_design\_feature=>  
 die\_clamping

#### 5.1.4.10.14 production\_die\_cast\_mould to die\_clamping (as stationary\_die\_items)

AIM element: PATH

Reference Path: production\_die\_cast\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die cast mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='stationary die clamping reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
die\_clamping

### 5.1.4.10.15 production\_die\_cast\_mould to shot\_sleeve (as stationary\_die\_items)

AIM element: PATH  
 Reference Path: production\_die\_cast\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production die cast mould occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='shot sleeve reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 die\_design\_feature=>  
 shot\_sleeve

### 5.1.4.11 Production\_die\_mould

AIM element: production\_die\_mould  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: production\_die\_mould<=  
 die\_design\_feature<=  
 production\_tool<=  
 casting\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.4.11.1 die\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_die\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'die type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

**5.1.4.11.2 production\_die\_mould to item\_size (as external\_box\_dimensions)**

AIM element: PATH

Reference Path: production\_die\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production die mould occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 feature\_component\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description = 'production die mould dimension usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_feature\_size

### 5.1.4.11.3 production\_die\_mould to slide (as slide\_requirement)

AIM element: PATH  
Reference Path: production\_die\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='slide reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
slide

**5.1.4.11.4 production\_die\_mould to core (as core\_for\_die)**

AIM element: PATH  
Reference Path: production\_die\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='slide reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
core

### 5.1.4.11.5 production\_die\_mould to die\_core (as core\_for\_die)

AIM element: PATH  
Reference Path: production\_die\_mould<=  
production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production die mould occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='slide reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
die\_core



### 5.1.4.11.6 production\_die\_mould to gating\_system (as gating)

AIM element: PATH  
 Reference Path: production\_die\_mould<=  
 production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'pattern plate occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='gating system reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 gating\_system

### 5.1.4.12 Production\_tool

AIM element: production\_tool  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: production\_tool<=  
 die\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.4.12.1 shrink\_factor

AIM element: length\_measure\_with\_unit  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature<=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <=  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <=  
property\_definition\_representation.definition  
{property\_definition\_representation=>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation=>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'shrink factor'}  
representation\_item=>  
measure\_representation\_item  
{measure\_representation\_item<=  
measure\_with\_unit=>  
length\_measure\_with\_unit}

### 5.1.4.12.2 male\_or\_female\_mould

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation =>  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'male or female'}  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {(descriptive\_representation\_item.description = 'male mould')  
 (descriptive\_representation\_item.description = 'female mould')}

### 5.1.4.12.3 number\_of\_cavities

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition <-  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation =>  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'number of cavities'}  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

#### 5.1.4.12.4 production\_tool to item\_size (as guide\_pin)

```

AIM element:      PATH
Reference Path:  production_tool<=
                 die_design_feature<=
                 casting_feature_definition<=
                 characterized_object
                 characterized_definition = characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'production mould occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 feature_component_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description = 'guide pin shape usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relating_shape_aspect ->
                 shape_aspect =>
                 casting_feature_size

```

### 5.1.4.12.5 production\_tool to die\_core (as core\_for\_die)

AIM element: PATH  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='die core reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relatng\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
die\_design\_feature=>  
die\_core

**5.1.4.12.6 production\_tool to ejector\_system (as ejection\_definition)**

AIM element: PATH

Reference Path: production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition=>  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production mould occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='ejector system reference usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 casting\_feature\_definition=>  
 ejector\_design\_feature=>  
 ejector\_system

#### 5.1.4.12.7 production\_tool to production\_core\_box (as core\_definition)

AIM element: PATH  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='production core box reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
sand\_mould\_design\_feature=>  
production\_core\_box

#### 5.1.4.12.8 production\_tool to machining\_allowance (as allowance\_on\_machinable\_surfaces)

AIM element: PATH  
Reference Path: production\_tool<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition= characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition=>  
machining\_allowance



**5.1.4.12.9 production\_tool to parting\_surface (as male\_or\_female\_parting)**

AIM element: PATH

Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)

Reference Path: production\_tool<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 {representation.name= 'female parting'}  
 shape\_representation=>  
 parting\_surface

### 5.1.4.12.10 production\_tool to parting\_surface (as offset\_parting\_surfaces)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_tool<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
{property\_definition=>  
product\_definition\_shape}  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
{representation.name= 'offset parting'}  
shape\_representation=>  
parting\_surface

**5.1.4.12.11 production\_tool to drafted\_surface (as draft)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: production\_tool<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation =>  
 shape\_representation=>  
 drafted\_surface

### 5.1.4.12.12 production\_tool to item\_size (as outside\_shape\_of\_die\_mould)

AIM element: PATH  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition<=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
feature\_component\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'production tool outside shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_feature\_size

**5.1.4.12.13 production\_tool to shape\_aspect (as outside\_shape\_of\_die\_mould)**

AIM element: PATH  
 Reference Path: production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production tool occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description=\n  
 'production tool outside shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect

#### 5.1.4.12.14 production\_tool to riser (as riser\_on\_mould)

AIM element: PATH  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='riser reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
riser

**5.1.4.12.15 production\_tool to riser\_contact (as riser\_on\_mould)**

```

AIM element:      PATH
Reference Path:  production_tool<=
                 die_design_feature<=
                 casting_feature_definition=>
                 characterized_object
                 characterized_definition=characterized_object
                 characterized_definition <-
                 property_definition.definition
                 property_definition =>
                 product_definition_shape <-
                 shape_aspect.of_shape
                 {shape_aspect
                 shape_aspect.description = 'production mould occurrence'}
                 shape_aspect <-
                 shape_aspect_relationship.related_shape_aspect
                 {[shape_aspect_relationship =>
                 shape_defining_relationship]
                 [shape_aspect_relationship
                 shape_aspect_relationship.description='riser contact reference usage']}
                 shape_aspect_relationship
                 shape_aspect_relationship.relating_shape_aspect ->
                 shape_aspect =>
                 casting_instanced_feature=>
                 casting_feature_definition=>
                 gating_design_feature=>
                 riser_contact

```

### 5.1.4.12.16 production\_tool to vent (as vent\_requirement)

AIM element: PATH  
Reference Path: production\_tool<=  
die\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'production mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='vent reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
vent



### 5.1.4.12.17 production\_tool to shape\_element (as mould\_impression\_shape)

AIM element: PATH  
 Reference Path: production\_tool<=  
 die\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <=  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'production tool occurrence'}  
 shape\_aspect <=  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description=\n  
 'production tool impression shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect

### 5.1.4.13 Secondary\_tooling

AIM element: secondary\_tooling  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: secondary\_tooling<=  
 die\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

### 5.1.4.13.1 type\_of\_tooling

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: secondary\_tooling<=  
die\_design\_feature  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'type of tooling'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.4.14 Shot\_sleeve

AIM element: shot\_sleeve  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: shot\_sleeve <=  
die\_design\_feature<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

#### 5.1.4.14.1 shot\_sleeve to item\_size (as shot\_sleeve\_size)

AIM element: PATH  
 Reference Path: shot\_sleeve<=  
   die\_design\_feature<=  
   casting\_feature\_definition<=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <=  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'shot sleeve occurrence'}  
   shape\_aspect <=  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   feature\_component\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description = 'shot sleeve shape usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relating\_shape\_aspect ->  
   shape\_aspect =>  
   casting\_feature\_size

#### 5.1.4.15 Slide

AIM element: slide  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
   subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: slide<=  
   die\_design\_feature<=  
   {casting\_feature\_definition =>  
   casting\_instanced\_feature}  
   casting\_feature\_definition <=  
   characterized\_object

### 5.1.4.15.1 slide\_activation\_method

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: slide<=  
die\_design\_feature  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'slide activation method'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.4.15.2 slide\_material

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: slide<=  
 die\_design\_feature  
 casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 material\_designation.definitions[i]  
 material\_designation  
 {(material\_designation.name = 'aluminum')  
 (material\_designation.name = 'plastic')  
 (material\_designation.name = 'cast iron')  
 (material\_designation.name = 'urethane')  
 (material\_designation.name = 'wood')}

### 5.1.4.15.3 slide to shape\_element (as shape)

AIM element: PATH  
 Reference Path: slide<=  
 die\_design\_feature<=  
 casting\_feature\_definition <=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'slide occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description='slide shape usage']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relatng\_shape\_aspect ->  
 shape\_aspect

## 5.1.5 Investment\_casting features UoF

### 5.1.5.1 Investment\_casting\_master

AIM element: investment\_casting\_master  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: investment\_casting\_master<=  
investment\_cast\_tooling<=  
casting\_production\_definition<=  
product\_definition<=  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

### 5.1.5.1.1 shrink\_factor\_for\_master\_tooling

AIM element: length\_measure\_with\_unit  
 Source: ISO 10303-41  
 Reference Path: production\_investment\_cast\_mould <=  
 investment\_design\_feature<=  
 casting\_feature\_definition=>  
 casting\_instanced\_feature <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation =>  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'shrink factor for master tooling'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.5.2 Investment\_casting\_tooling

AIM element: investment\_cast\_tooling  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: investment\_cast\_tooling<=  
 casting\_production\_definition<=  
 product\_definition<=  
 product\_definition.frame\_of\_reference->  
 product\_definition\_context<=  
 application\_context\_element  
 application\_context\_element.name='functional definition'

### 5.1.5.2.1 tooling\_identifier

AIM element: descriptive\_representation\_item  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: investment\_cast\_tooling<=  
casting\_production\_definition  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition<=  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'tooling identifier'}  
representation\_item =>  
descriptive\_representation\_item



### 5.1.5.2.2 investment\_casting\_tooling to material (as tooling\_material)

AIM element: material\_designation.name  
 Source: ISO 10303-45  
 Reference Path: investment\_cast\_tooling<=  
 casting\_production\_definition  
 product\_definition  
 product\_definition\_relationship.relying\_product\_definition  
 {product\_definition\_relationship.description='raw material'  
 product\_definition\_relationship =>  
 product\_definition\_usage =>  
 make\_from\_usage\_option}  
 product\_definition\_relationship  
 product\_definition\_relationship.related\_product\_definition ->  
 product\_definition  
 {product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 material\_designation.definitions[i]  
 material\_designation}  
 material\_designation.name  
 {(material\_designation.name = 'plaster')  
 (material\_designation.name = 'expandable polystyrene')  
 (material\_designation.name = 'iron')  
 (material\_designation.name = 'soft wood')  
 (material\_designation.name = 'shell')  
 (material\_designation.name = 'aluminum')  
 (material\_designation.name = 'epoxy')  
 (material\_designation.name = 'wood metal reinforced')  
 (material\_designation.name = 'brass')  
 (material\_designation.name = 'hard wood')  
 (material\_designation.name = 'urethane')  
 (material\_designation.name = 'other')  
 (material\_designation.name = 'steel')  
 (material\_designation.name = 'ceramic')  
 (material\_designation.name = 'polymethylmethacrylate')  
 (material\_designation.name = 'plywood')} }

### 5.1.5.2.3 investment\_casting\_tooling to investment\_casting\_tooling\_definition (as die\_equipment)

AIM element: PATH  
Reference Path: investment\_cast\_tooling<=  
casting\_production\_definition<=  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n'investment casting tooling definition reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relatng\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
investment\_design\_feature=>  
investment\_casting\_tooling\_definition

### 5.1.5.2.4 investment\_casting\_tooling to cast\_part\_with\_rigging (as prototype)

AIM element: PATH  
Reference Path: investment\_cast\_tooling<=  
casting\_production\_definition<=  
product\_definition<-  
product\_definition\_relationship.relatng\_product\_definition  
product\_definition\_relationship  
product\_definition\_relationship.related\_product\_definition->  
product\_definition=>  
casting\_production\_definition=>  
cast\_part\_with\_rigging

### 5.1.5.3 Investment\_casting\_tooling\_definition

AIM element: investment\_casting\_tooling\_definition  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: investment\_casting\_tooling\_definition<=  
 investment\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.5.3.1 investment\_casting\_tooling\_definition to investment\_mould (as tree\_structure)

AIM element: PATH  
 Reference Path: investment\_casting\_tooling\_definition <=  
 investment\_design\_feature<=  
 casting\_feature\_definition =>  
 characterized\_object  
 characterized\_definition= characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape <-  
 shape\_aspect.of\_shape  
 {shape\_aspect  
 shape\_aspect.description = 'investment casting tooling definition occurrence'}  
 shape\_aspect <-  
 shape\_aspect\_relationship.related\_shape\_aspect  
 {[shape\_aspect\_relationship =>  
 shape\_defining\_relationship]  
 [shape\_aspect\_relationship  
 shape\_aspect\_relationship.description=\  
 'investment casting tooling definition occurrence']}  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relate\_shape\_aspect ->  
 shape\_aspect =>  
 casting\_instanced\_feature=>  
 investment\_design\_definition=>  
 investment\_mould

### 5.1.5.3.2 investment\_casting\_tooling\_definition to production\_investment\_cast\_mould (as die\_mould)

AIM element: PATH

Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling definition occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n'production investment cast mould reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
investment\_design\_feature=>  
production\_investment\_cast\_mould

### 5.1.5.3.3 investment\_casting\_tooling\_definition to lost\_foam\_casting\_die (as die\_mould)

AIM element: PATH

Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_feature\_definition=>  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling definition occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n  
'production investment cast mould reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
investment\_design\_feature=>  
lost\_foam\_casting\_die

#### 5.1.5.3.4 investment\_casting\_tooling\_definition to sprue\_and\_runner\_mould (as metal\_flow\_system)

AIM element: PATH

Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition= characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n'sprue and runner mould reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
instanced\_feature=>  
casting\_feature\_definition=>  
investment\_design\_feature=>  
sprue\_and\_runner\_mould

### 5.1.5.3.5 investment\_casting\_tooling\_definition to production\_core\_box (as core\_tooling)

AIM element: PATH

Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition= characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling definition occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description=\n  
'production core box reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
production\_core\_box

### 5.1.5.3.6 investment\_casting\_tooling\_definition to assembly (as pattern\_section\_assembly))

AIM element: PATH  
Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_instanced\_feature =>  
shape\_aspect<-  
applied\_action\_method\_assignment.items[i]  
applied\_action\_method\_assignment<=  
action\_method\_assignment  
action\_method\_assignment.assigned\_action\_method->  
action\_method=>  
casting\_activity=>  
non\_casting\_activity=>  
assembly

### 5.1.5.3.7 investment\_casting\_tooling\_definition to shape\_element (as pattern\_section\_definition)

AIM element: PATH  
Reference Path: investment\_casting\_tooling\_definition<=  
investment\_design\_feature<=  
casting\_feature\_definition =>  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'investment casting tooling definition occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'pattern section shape usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect



### 5.1.5.4 Investment\_design\_feature

AIM element: investment\_design\_feature  
 Source: ISO 10303-223  
 Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
 subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
 Reference Path: investment\_design\_feature<=  
 {casting\_feature\_definition =>  
 casting\_instanced\_feature}  
 casting\_feature\_definition <=  
 characterized\_object

#### 5.1.5.4.1 investment\_design\_feature to orientation (as placement)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: investment\_design\_feature<=  
 casting\_feature\_definition<=  
 characterized\_object  
 characterized\_definition = characterized\_object  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 {property\_definition=>  
 product\_definition\_shape}  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'orientation'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.5.5 Investment\_mould

AIM element: investment\_mould  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: investment\_mould <=  
investment\_design\_feature  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

### 5.1.5.5.1 number\_of\_patterns

AIM element: PATH

Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)

Reference Path: investment\_mould<=  
investment\_design\_feature<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'number of patterns'}  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit  
{measure\_with\_unit.unit\_component->  
unit=named\_unit  
named\_unit=>  
context\_dependent\_unit  
context\_dependent\_unit.name='count'}  
measure\_with\_unit.value\_component ->  
measure\_value  
measure\_value = count\_measure  
count\_measure

### 5.1.5.6 Production\_investment\_cast\_mould

AIM element: production\_investment\_cast\_mould  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: production\_investment\_cast\_mould <=  
investment\_design\_feature<=  
production\_tool<=  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

#### 5.1.5.6.1 production\_investment\_cast\_mould to in\_facility\_location (as die\_mould\_location)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: production\_investment\_cast\_mould <=  
investment\_design\_feature<=  
production\_tool<=  
casting\_feature\_definition=>  
casting\_instanced\_feature <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
{property\_definition=>  
product\_definition\_shape}  
property\_definition <-  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation =>  
in\_facility\_location

### 5.1.5.6.2 production\_investment\_cast\_mould to person\_and\_organization (as die\_mould\_location)

AIM element: PATH  
 Reference Path: die\_mould\_tooling<=  
   casting\_production\_definition  
   product\_definition  
   person\_and\_organization\_item = product\_definition  
   person\_and\_organization\_item<-  
   applied\_person\_and\_organization\_assignment.items[i]  
   applied\_person\_and\_organization\_assignment<=  
   person\_and\_organization\_assignment  
   person\_and\_organization\_assignment.assignment\_person\_and\_organization->  
   person\_and\_organization

### 5.1.5.6.3 production\_investment\_cast\_mould to slide (as slide\_requirement)

AIM element: PATH  
 Reference Path: production\_investment\_cast\_mould <=  
   investment\_design\_feature<=  
   production\_tool<=  
   casting\_feature\_definition <=  
   characterized\_object  
   characterized\_definition = characterized\_object  
   characterized\_definition <-  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape <-  
   shape\_aspect.of\_shape  
   {shape\_aspect  
   shape\_aspect.description = 'production investment cast mould occurrence'}  
   shape\_aspect <-  
   shape\_aspect\_relationship.related\_shape\_aspect  
   {[shape\_aspect\_relationship =>  
   shape\_defining\_relationship]  
   [shape\_aspect\_relationship  
   shape\_aspect\_relationship.description='slide reference usage']}  
   shape\_aspect\_relationship  
   shape\_aspect\_relationship.relatng\_shape\_aspect ->  
   shape\_aspect  
   casting\_instanced\_feature=>  
   casting\_feature\_definition=>  
   die\_design\_feature=>  
   slide

### 5.1.5.7 Sprue\_and\_runner\_mould

AIM element: sprue\_and\_runner\_mould  
Source: ISO 10303-223  
Rules: casting\_feature\_life\_cycle – (See 5.2.4.2)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference Path: sprue\_and\_runner\_mould<=  
investment\_design\_feature  
{casting\_feature\_definition =>  
casting\_instanced\_feature}  
casting\_feature\_definition <=  
characterized\_object

#### 5.1.5.7.1 sprue\_and\_runner\_mould to runner (as gate\_runner)

AIM element: PATH  
Reference Path: sprue\_and\_runner\_mould<=  
investment\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue and runner mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='runner reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
runner

### 5.1.5.7.2 sprue\_and\_runner\_mould to sprue (as pouring\_sprue)

AIM element: PATH

Reference Path: sprue\_and\_runner\_mould  
investment\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue and runner mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='sprue reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
sprue

### 5.1.5.7.3 sprue\_and\_runner\_mould to ingate (as rigging\_component)

AIM element: PATH  
Reference Path: sprue\_and\_runner\_mould  
investment\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue and runner mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='ingate reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature\_definition=>  
gating\_design\_feature=>  
ingate



#### 5.1.5.7.4 sprue\_and\_runner\_mould to investment\_mould (as mould\_assembly)

AIM element: PATH

Reference Path: sprue\_and\_runner\_mould  
investment\_design\_feature<=  
casting\_feature\_definition <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'sprue and runner mould occurrence'}  
shape\_aspect <=  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description='ingate reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect =>  
casting\_instanced\_feature=>  
investment\_design\_definition=>  
investment\_mould

ISO 10303-223:2008 (E)

## 5.1.6 Library\_reference UoF

### 5.1.6.1 BSU

#1: BSU is a Class\_BSU

AIM element: externally\_defined\_class  
Source: ISO 10303-223

#2: BSU is a Property\_BSU

AIM element: externally\_defined\_general\_property  
Source: ISO 10303-223

#3: BSU is a Supplier\_BSU

AIM element: organization  
Source: ISO 10303-41

#### 5.1.6.1.1 code

#1: BSU is a Class\_BSU

AIM element: externally\_defined\_item.item\_id  
Source: ISO 10303-41  
Reference Path: externally\_defined\_class <=  
externally\_defined\_item  
externally\_defined\_item.item\_id

#2: BSU is a Property\_BSU

AIM element: externally\_defined\_item.item\_id  
Source: ISO 10303-41  
Reference Path: externally\_defined\_general\_property <=  
externally\_defined\_item  
externally\_defined\_item.item\_id

#3: BSU is a Supplier\_BSU

AIM element: organization.id  
Source: ISO 10303-41  
Reference Path: organization.id

### 5.1.6.2 Class\_BSU

AIM element: externally\_defined\_class  
 Source: ISO 10303-223  
 Rules: restrict\_name\_for\_known\_source – (See 5.2.4.29)  
 externally\_defined\_class\_with\_known\_source\_requirement – (See 5.2.4.17)  
 Reference Path: [externally\_defined\_class <=  
 class <=  
 group]  
 [externally\_defined\_class <=  
 externally\_defined\_item  
 {externally\_defined\_item.source ->  
 external\_source =>  
 known\_source <=  
 pre\_defined\_item  
 pre\_defined\_item.name = 'ISO 13584 library'}]

#### 5.1.6.2.1 version

AIM element: identification\_assignment.assigned\_id  
 Source: ISO 10303-41  
 Reference Path: externally\_defined\_class  
 externally\_defined\_class = external\_identification\_item  
 external\_identification\_item<-  
 applied\_external\_identification\_assignment.items[i]  
 {applied\_external\_identification\_assignment =>  
 library\_class\_version\_assignment}  
 applied\_external\_identification\_assignment <=  
 external\_identification\_assignment<=  
 {identification\_assignment.role ->  
 identification\_role  
 [identification\_role.name = 'class version']}  
 identification\_assignment  
 identification\_assignment.assigned\_id

### 5.1.6.2.2 class\_BSU to supplier\_BSU (as defined\_by)

AIM element: organization  
Source: ISO 10303-41  
Rule:  
Reference Path: externally\_defined\_class <=  
externally\_defined\_item  
externally\_defined\_item.source ->  
external\_source =>  
known\_source  
organization\_item = known\_source  
organization\_item <-  
applied\_organization\_assignment.items[i]  
applied\_organization\_assignment <=  
organization\_assignment  
{organization\_assignment.role ->  
organization\_role  
[organization\_role.name = 'library supplier']}  
organization\_assignment  
organization\_assignment.assigned\_organization ->  
organization

### 5.1.6.3 Externally\_defined\_representation

AIM element: externally\_defined\_representation\_with\_parameters  
Source: ISO 10303-223  
Reference Path: externally\_defined\_representation\_with\_parameters<=  
representation

#### 5.1.6.3.1 placement

AIM element: placement  
Source: ISO 10303-42  
Reference Path: externally\_defined\_representation\_with\_parameters<=  
representation  
representation.items[i]->  
representation\_item=>  
geometric\_representation\_item=>  
placement

**5.1.6.3.2 location**

AIM element: placement.location  
 Source: ISO 10303-42  
 Reference Path: externally\_defined\_representation\_with\_parameters<=  
 Representation  
 representation.items[i]->  
 representation\_item=>  
 geometric\_representation\_item=>  
 placement  
 placement.location

**5.1.6.3.3 externally\_defined\_representation to library\_part\_assignment (as identified\_by)**

AIM element: PATH  
 Reference Path: externally\_defined\_representation\_with\_parameters  
 classification\_item=externally\_defined\_representation\_with\_parameters<-  
 applied\_classification\_assignment.items[i]  
 applied\_classification\_assignment

**5.1.6.4 Library\_part\_assignment**

AIM element: applied\_classification\_assignment  
 Source: ISO 10303-223  
 Reference Path: applied\_classification\_assignment<=  
 classification\_assignment

**5.1.6.4.1 library\_part\_assignment to class\_BSU (as definitional\_class\_BSU)**

AIM element: PATH  
 Reference Path: applied\_classification\_assignment <=  
 classification\_assignment  
 {classification\_assignment.role ->  
 classification\_role  
 classification\_role.name = 'definitional class membership'}  
 classification\_assignment.assigned\_class ->  
 group =>  
 {group.name = 'library identifier'}  
 class =>  
 externally\_defined\_class

### 5.1.6.4.2 library\_part\_assignment to property\_value (as definitional\_property\_value\_pairs)

AIM element: PATH  
Reference Path: applied\_classification\_assignment <=  
classification\_assignment  
{classification\_assignment.role ->  
classification\_role  
classification\_role.name = 'definitional class membership'}  
classification\_assignment.assigned\_class ->  
group =>  
{group.name = 'library identifier'}  
class =>  
externally\_defined\_class <=  
externally\_defined\_item <-  
externally\_defined\_item\_relationship.related\_item  
{externally\_defined\_item\_relationship.name = 'name scope'}  
externally\_defined\_item\_relationship  
externally\_defined\_item\_relationship.relating\_item->  
externally\_defined\_item =>  
externally\_defined\_general\_property <=  
general\_property <-  
general\_property\_association.base\_definition  
general\_property\_association  
{general\_property\_association.name = 'definitional'}  
general\_property\_association.derived\_definition->  
property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation  
{representation.name = 'property value'}

### 5.1.6.5 Property\_BSU

AIM element: externally\_defined\_general\_property  
 Source: ISO 10303-223  
 Rules: restrict\_name\_for\_known\_source – (See 5.2.4.29)  
 Reference Path: [externally\_defined\_general\_property <=  
 general\_property]  
 [externally\_defined\_general\_property <=  
 externally\_defined\_item  
 {externally\_defined\_item.source ->  
 external\_source =>  
 known\_source <=  
 pre\_defined\_item  
 pre\_defined\_item.name = 'ISO 13584 library'}]

#### 5.1.6.5.1 version

AIM element: identification\_assignment.assigned\_id  
 Source: ISO 10303-41  
 Reference Path: externally\_defined\_general\_property  
 externally\_defined\_general\_property = external\_identification\_item  
 external\_identification\_item<-  
 applied\_external\_identification\_assignment.items[i]  
 {applied\_external\_identification\_assignment =>  
 library\_class\_version\_assignment}  
 applied\_external\_identification\_assignment <=  
 external\_identification\_assignment<=  
 identification\_assignment  
 {identification\_assignment.role->  
 identification\_role  
 [identification\_role.name = 'property\_version']}  
 identification\_assignment.assigned\_id

#### 5.1.6.5.2 property\_BSU to class\_BSU (as name\_scope)

AIM element: PATH  
 Reference Path: externally\_defined\_general\_property <=  
 externally\_defined\_item <-  
 externally\_defined\_item\_relationship.relying\_item  
 externally\_defined\_item\_relationship  
 {externally\_defined\_item\_relationship.name = 'name scope'}  
 externally\_defined\_item\_relationship.related\_item ->  
 externally\_defined\_item =>  
 externally\_defined\_class

### 5.1.6.6 Property\_value

AIM element: representation  
Source: ISO 10303-43  
Reference Path: representation  
{representation.name = 'property value'}

#### 5.1.6.6.1 property\_value to property\_BSU (as property\_BSU)

AIM element: PATH  
Reference Path: representation <-  
property\_definition\_representation.used\_representation  
property\_definition\_representation  
property\_definition\_representation.definition ->  
represented\_definition=general\_property  
general\_property =>  
externally\_defined\_general\_property

#### 5.1.6.6.2 value\_amount

#1: boolean

AIM element: descriptive\_representation\_item  
Source: ISO 10303-45  
Reference Path: representation  
representation.items[i] ->  
representation\_item =>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'TRUE')  
(descriptive\_representation\_item.description = 'FALSE')}

#2: integer

AIM element: value\_representation\_item  
Source: ISO 10303-43  
Reference Path: representation  
representation.items[i] ->  
representation\_item =>  
value\_representation\_item  
value\_representation\_item.value\_component  
{value\_representation\_item.value\_component->  
measure\_value  
measure\_value=count\_measure}



## #3: logical

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-45  
 Reference Path: representation  
 representation.items[i] ->  
 representation\_item =>  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'TRUE')  
 (descriptive\_representation\_item.description = 'FALSE')  
 (descriptive\_representation\_item.description = 'UNKNOWN')}

## #4: number

AIM element: value\_representation\_item  
 Source: ISO 10303-43  
 Reference Path: representation  
 representation.items[i] ->  
 representation\_item =>  
 value\_representation\_item  
 value\_representation\_item.value\_component  
 {value\_representation\_item.value\_component->  
 measure\_value  
 measure\_value=count\_measure}

## #5: real

AIM element: value\_representation\_item  
 Source: ISO 10303-43  
 Reference Path: representation  
 representation.items[i] ->  
 representation\_item =>  
 value\_representation\_item  
 value\_representation\_item.value\_component  
 {value\_representation\_item.value\_component->  
 measure\_value  
 measure\_value=length\_measure}

## #6: string

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-45  
 Reference Path: representation  
 representation.items[i] ->  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description

### 5.1.6.7 Supplier\_bsu

AIM element: organization  
Source: ISO 10303-41  
Rules:  
Reference Path: externally\_defined\_class <=  
externally\_defined\_item  
externally\_defined\_item.source ->  
external\_source =>  
known\_source  
organization\_item = known\_source  
organization\_item <-  
applied\_organization\_assignment.items[i]  
applied\_organization\_assignment <=  
organization\_assignment  
{organization\_assignment.role ->  
organization\_role  
[organization\_role.name = 'library supplier']}  
organization\_assignment  
organization\_assignment.assigned\_organization ->  
organization

## 5.1.7 Manufacturing\_casting\_resources UoF

### 5.1.7.1 Bankout\_frame

AIM element: bankout\_frame  
Source: ISO 10303-223  
Rule:  
Reference Path: bankout\_frame  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

### 5.1.7.1.1 material\_type

AIM element: PATH  
 Reference Path: bankout\_frame<=  
 core\_equipment<=  
 casting\_equipment<=  
 machine<=  
 action\_resource  
 characterized\_resource\_definition= action\_resource  
 characterized\_resource\_definition<-  
 resource\_property.resource  
 resource\_property <-  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation  
 representation.name='material type'  
 representation.items [i]->  
 representation\_item=>  
 descriptive\_representation\_item

### 5.1.7.2 Controller

AIM element: controller  
 Source: ISO 10303-223  
 Rule:  
 Reference Path: controller<=  
 action\_resource

#### 5.1.7.2.1 company\_model

AIM element: action\_resource.description  
 Source: ISO 10303-41  
 Reference Path: controller<=  
 action\_resource  
 action\_resource.description

#### 5.1.7.2.2 company\_name

AIM element: action\_resource.name  
 Source: ISO 10303-41  
 Reference Path: controller<=  
 action\_resource  
 action\_resource.name

### 5.1.7.2.3 controller to specification (as controller\_specification)

AIM element: PATH  
Reference Path: controller  
document\_reference\_item=controller  
document\_reference\_item<-  
applied\_document\_reference.items [i]  
applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document=>  
process\_plan\_specification

### 5.1.7.3 Core\_bench

AIM element: core\_bench  
Source: ISO 10303-223  
Reference Path: core\_bench<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

### 5.1.7.3.1 core\_required

AIM element: PATH

Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)

Reference Path: core\_bench<=  
 core\_equipment<=  
 casting\_equipment<=  
 machine<=  
 action\_resource  
 characterized\_resource\_definition= action\_resource  
 characterized\_resource\_definition<-  
 resource\_property.resource  
 resource\_property <-  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation =>  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'core required'}  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='count'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

### 5.1.7.3.2 wax\_vent

AIM element: PATH  
Reference Path: core\_bench<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource  
characterized\_resource\_definition= action\_resource  
characterized\_resource\_definition<-  
resource\_property.resource  
resource\_property <-  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.name='wax vent'  
representation.items [i]->  
representation\_item=>  
descriptive\_representation\_item

### 5.1.7.4 Core\_box\_vent

AIM element: core\_box\_vent  
Source: ISO 10303-223  
Reference path: core\_box\_vent<=  
resource\_property

#### 5.1.7.4.1 material\_type

AIM element: material\_designation.name  
Source: ISO 10303-45  
Reference Path: core\_box\_vent<=  
resource\_property<-  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.name='matgerial type'  
representation.items [i]->  
representation\_item=>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'nylon')  
(descriptive\_representation\_itemn.description = 'brass')  
(descriptive\_representation\_item.description = 'stainless steel')  
(descriptive\_representation\_item.description = 'polyurethane')};

### 5.1.7.4.2 vent\_type

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: core\_box\_vent<=  
   resource\_property  
   resource\_property <-  
   resource\_property\_representation.property  
   resource\_property\_representation  
   resource\_property\_representation.representation ->  
   representation  
   representation.items[i] ->  
   {representation\_item  
   representation\_item.name = 'vent type'}  
   representation\_item =>  
   descriptive\_representation\_item  
   descriptive\_representation\_item.description

### 5.1.7.4.3 core\_box\_vent to equipment\_size (as vent\_dimensions)

AIM element: PATH  
 Reference Path: core\_box\_vent<=  
   resource\_property  
   resource\_property <-  
   resource\_property\_representation.property  
   resource\_property\_representation  
   resource\_property\_representation.representation ->  
   representation

### 5.1.7.4.4 core\_box\_vent to shape\_element (as core\_shape)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: core\_box\_vent<=  
   resource\_property  
   resource\_property <-  
   resource\_property\_representation.property  
   resource\_property\_representation  
   resource\_property\_representation.representation ->  
   representation=>  
   shape\_representation

### 5.1.7.5 Core\_conveyer\_system

AIM element: core\_conveyer\_system  
Source: ISO 10303-223  
Reference Path: core\_conveyer\_system<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

#### 5.1.7.5.1 cores\_required

AIM element: PATH  
Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
Reference Path: core\_conveyer\_system<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource  
characterized\_resource\_definition= action\_resource  
characterized\_resource\_definition<-  
resource\_property.resource  
resource\_property <-  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation =>  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'core required'}  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit  
{measure\_with\_unit.unit\_component->  
unit=named\_unit  
named\_unit=>  
context\_dependent\_unit  
context\_dependent\_unit.name='count'}  
measure\_with\_unit.value\_component ->  
measure\_value  
measure\_value = count\_measure  
count\_measure



### 5.1.7.5.2 cycle\_time

AIM element: PATH  
 Reference Path: core\_conveyer\_system<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   characterized\_resource\_definition= action\_resource  
                   characterized\_resource\_definition<-  
                   resource\_property.resource  
                   resource\_property <-  
                   resource\_property\_representation.property  
                   resource\_property\_representation  
                   resource\_property\_representation.representation ->  
                   representation  
                   representation.name='cycle time'  
                   representation.items [i]->  
                   representation\_item=>  
                   descriptive\_representation\_item

### 5.1.7.5.3 core\_conveyer\_system\_to\_equipment\_size (as core\_box\_skid)

AIM element: PATH  
 Reference Path: core\_conveyer\_system<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   characterized\_resource\_definition= action\_resource  
                   characterized\_resource\_definition<-  
                   resource\_property.resource  
                   resource\_property <-  
                   resource\_property\_representation.property  
                   resource\_property\_representation  
                   resource\_property\_representation.representation ->  
                   representation

#### 5.1.7.5.4 core\_conveyer\_system to equipment\_size (as core\_plug)

AIM element: PATH  
Reference Path: core\_conveyer\_system<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource  
characterized\_resource\_definition= action\_resource  
characterized\_resource\_definition<=  
resource\_property.resource  
resource\_property <=  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation

#### 5.1.7.6 Core\_equipment

AIM element: core\_equipment  
Source: ISO 10303-223  
Reference Path: core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

#### 5.1.7.7 Core\_machine

AIM element: core\_machine  
Source: ISO 10303-223  
Reference Path: core\_machine  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

**5.1.7.7.1 core\_machine to core\_box\_vent (as vent definition)**

AIM element: PATH  
 Reference Path: core\_machine<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   characterized\_resource\_definition= action\_resource  
                   characterized\_resource\_definition<-  
                   resource\_property.resource  
                   resource\_property =>  
                   core\_box\_vent

**5.1.7.7.2 core\_machine to ejector\_equipment(as ejector\_definition)**

AIM element: PATH  
 Reference Path: core\_machine<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   action\_resource\_relationship.related\_resource  
                   action\_resource\_relationship  
                   {action\_resource\_relationship=>  
                   machine\_element\_relationship}  
                   action\_resource\_relationship.relying\_resource->  
                   action\_resource=>  
                   machine=>  
                   casting\_equipment=>  
                   ejector\_equipment

**5.1.7.7.3 core\_machine to machine\_mounting (as mounting\_definition)**

AIM element: PATH  
 Reference Path: core\_machine<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   characterized\_resource\_definition= action\_resource  
                   characterized\_resource\_definition<-  
                   resource\_property.resource  
                   resource\_property =>  
                   machine\_mounting

### 5.1.7.8 Core\_setter

AIM element: core\_setter  
Source: ISO 10303-223  
Reference Path: core\_setter<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

#### 5.1.7.8.1 moulding\_machine

AIM element: PATH  
Reference Path: core\_setter<=  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource  
characterized\_resource\_definition= action\_resource  
characterized\_resource\_definition<-  
resource\_property.resource  
resource\_property <-  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.name='moulding machine'  
representation.items [i]->  
representation\_item=>  
descriptive\_representation\_item

### 5.1.7.9 Dryer

AIM element: dryer  
Source: ISO 10303-223  
Reference Path: dryer  
core\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

### 5.1.7.9.1 material\_type

AIM element: PATH  
 Reference Path: dryer<=  
                   core\_equipment<=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource  
                   characterized\_resource\_definition= action\_resource  
                   characterized\_resource\_definition<-  
                   resource\_property.resource  
                   resource\_property <-  
                   resource\_property\_representation.property  
                   resource\_property\_representation  
                   resource\_property\_representation.representation ->  
                   representation  
                   representation.name='material type'  
                   representation.items [i]->  
                   representation\_item=>  
                   descriptive\_representation\_item

### 5.1.7.10 Ejector\_equipment

AIM element: ejector\_equipment  
 Source: ISO 10303-223  
 Reference Path: ejector\_equipment <=  
                   casting\_equipment<=  
                   machine<=  
                   action\_resource

### 5.1.7.10.1 ejector\_equipment to ejector\_system(as ejector\_characteristics)

AIM element: PATH  
Reference Path: ejector\_equipment <=  
casting\_equipment<=  
machine<=  
action\_resource  
action\_resource.usage[i]->  
supported\_item  
supported\_item=action\_method  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment =>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i] ->  
action\_item  
action\_item = shape\_aspect  
shape\_aspect  
casting\_instanced\_feature=>  
casting\_feature definition=>  
ejector\_design\_feature=>  
ejector\_system<=

### 5.1.7.11 Equipment

AIM element: casting\_equipment  
Source: ISO 10303-223  
Reference Path: casting\_equipment<=  
machine<=  
action\_resource

### 5.1.7.11.1 equipment\_model

AIM element: action\_resource.description  
Source: ISO 10303-41  
Reference Path: casting\_equipment<=  
machine<=  
action\_resource  
action\_resource.description

### 5.1.7.11.2 equipment\_name

AIM element: organization.name  
 Source: ISO 10303-41  
 Reference Path: casting\_equipment<=  
 machine<=  
 action\_resource  
 organization\_item=action\_resource  
 organization\_item <=  
 applied\_organization\_assignment.items [i]<=  
 applied\_organization\_assignment <=  
 organization\_assignment  
 {organization\_assignment.role->  
 organization\_role  
 organization\_role.name='equipment company name'}  
 organization\_assignment.assigned\_organization ->  
 organization  
 organization.name

### 5.1.7.11.3 equipment\_type

AIM element: action\_resource\_type.name  
 Source: ISO 10303-41  
 Reference Path: casting\_equipment<=  
 machine<=  
 action\_resource  
 characterized\_resource\_definition=action\_resource  
 characterized\_resource\_definition  
 resource\_property.resource->  
 resource\_property  
 resource\_property <=  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation  
 representation.name='equipment type'  
 representation.items [i]->  
 representation\_item=>  
 descriptive\_representation\_item

#### 5.1.7.11.4 identification

AIM element: action\_resource.name  
Source: ISO 10303-41  
Reference Path: casting\_equipment<=  
machine<=  
action\_resource  
action\_resource.name

#### 5.1.7.11.5 equipment to in\_facility\_location (as location)

AIM element: PATH  
Reference Path: casting\_equipment<=  
machine<=  
action\_resource  
characterized\_resource\_definition=action\_resource  
characterized\_resource\_definition  
resource\_property.resource->  
resource\_property  
resource\_property <-  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation=>  
in\_facility\_location

#### 5.1.7.11.6 equipment to controller (as controlled\_by)

AIM element: PATH  
Reference Path: casting\_equipment<=  
machine<=  
action\_resource<-  
action\_resource\_relationship.relating\_resource  
action\_resource\_relationship =>  
machine\_element\_relationship  
action\_resource\_relationship.related\_resource->  
action\_resource=>  
controller

#### 5.1.7.12 Equipment\_size

AIM element: representation  
Source: ISO 10303-43  
Reference Path: representation  
representation.name = 'equipment\_size'



**5.1.7.12.1 equipment\_size to numeric\_parameter (as item\_height)**

AIM element: PATH  
 Reference Path: representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'height'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

**5.1.7.12.2 equipment\_size to numeric\_parameter (as item\_length)**

AIM element: PATH  
 Reference Path: representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'length'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

**5.1.7.12.3 equipment\_size to numeric\_parameter (as item\_width)**

AIM element: PATH  
 Reference Path: representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'width'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

#### 5.1.7.12.4 equipment\_size to numeric\_parameter (as item\_diameter)

AIM element: PATH  
Reference Path: representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'diameter'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

#### 5.1.7.12.5 equipment\_size to numeric\_parameter (as item\_draft)

AIM element: PATH  
Reference Path: representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'draft'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

#### 5.1.7.13 Finishing

AIM element: finishing\_equipment  
Source: ISO 10303-223  
Reference Path: finishing\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

#### 5.1.7.14 Generic\_casting\_resource

AIM element: generic\_manufacturing\_resource  
Source: ISO 10303-223  
Reference Path: generic\_manufacturing\_resource<=  
action\_resource

### 5.1.7.14.1 description

AIM element: action\_resource.description  
 Source: ISO 10303-41  
 Reference Path: generic\_manufacturing\_resource<=  
 action\_resource  
 action\_resource.description

### 5.1.7.14.2 name

AIM element: action\_resource.name  
 Source: ISO 10303-41  
 Reference Path: generic\_manufacturing\_resource<=  
 action\_resource  
 action\_resource.name

### 5.1.7.14.3 quantity

AIM element: count\_measure  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: generic\_manufacturing\_resource<=  
 action\_resource  
 characterized\_resource\_definition=action\_resource  
 characterized\_resource\_definition <-  
 resource\_property.resource  
 resource\_property <-  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation  
 representation.items [i]->  
 representation\_item =>  
 {representation\_item.name='resource quantity'}  
 measure\_representation\_item <=  
 measure\_with\_unit  
 {measure\_with\_unit.unit\_component->  
 unit=named\_unit  
 named\_unit=>  
 context\_dependent\_unit  
 context\_dependent\_unit.name='each'}  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

#### 5.1.7.14.4 units

AIM element: context\_dependent\_unit  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
Reference Path: generic\_manufacturing\_resource <=  
action\_resource  
characterized\_resource\_definition=action\_resource  
characterized\_resource\_definition <-  
resource\_property.resource  
resource\_property <-  
{resource\_property.name='quantity'}  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.unit\_component ->  
unit  
unit=named\_unit  
named\_unit <=  
context\_dependent\_unit  
{context\_dependent\_unit.name='each'}

#### 5.1.7.15 In\_facility\_location

AIM element: in\_facility\_location  
Source: ISO 10303-223  
Reference Path: in\_facility\_location <=  
representation

##### 5.1.7.15.1 building\_or\_area

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: in\_facility\_location <=  
representation  
{representation.name='building or area'}  
representation.items[i] ->  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

**5.1.7.15.2 location\_code**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: in\_facility\_location<=  
 representation  
 {representation.name = 'location code'}  
 representation.items[i] ->  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description

**5.1.7.15.3 sublocation**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: in\_facility\_location<=  
 representation  
 {representation.name = 'sublocation'}  
 representation.items[i] ->  
 representation\_item =>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description

**5.1.7.16 Machine\_mounting**

AIM element: machine\_mounting  
 Source: ISO 10303-223  
 Reference path: machine\_mounting<=  
 resource\_property

### 5.1.7.16.1 machine\_mounting\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: machine\_mounting<=  
action\_resource  
characterized\_resource\_definition=action\_resource  
characterized\_resource\_definition  
resource\_property.resource->  
resource\_property  
resource\_property <=  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'machine mounting type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.7.16.2 material\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: machine\_mounting<=  
action\_resource  
characterized\_resource\_definition=action\_resource  
characterized\_resource\_definition  
resource\_property.resource->  
resource\_property  
resource\_property <=  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'material type'}  
representation\_item =>  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.7.16.3 machine\_mounting to equipment\_size (as dimensions)

AIM element: PATH  
 Reference Path: machine\_mounting<=  
 action\_resource  
 characterized\_resource\_definition=action\_resource  
 characterized\_resource\_definition  
 resource\_property.resource->  
 resource\_property  
 resource\_property <=  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation

### 5.1.7.16.4 machine\_mounting to location\_element (as location)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: machine\_mounting<=  
 action\_resource  
 characterized\_resource\_definition=action\_resource  
 characterized\_resource\_definition  
 resource\_property.resource->  
 resource\_property  
 resource\_property <=  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation  
 shape\_representation=>  
 location\_shape\_representation

### 5.1.7.17 Melting\_equipment

AIM element: melting\_equipment  
 Source: ISO 10303-223  
 Reference Path: melting\_equipment<=  
 casting\_equipment<=  
 machine<=  
 action\_resource

### 5.1.7.18 Moulding\_equipment

AIM element: moulding\_equipment  
Source: ISO 10303-223  
Reference Path: moulding\_equipment<=  
casting\_equipment<=  
machine<=  
action\_resource

### 5.1.7.19 Resource\_with\_material

AIM element: resource\_with\_material  
Source: ISO 10303-223  
Reference Path: resource\_with\_material<=  
generic\_manufacturing\_resource<=  
action\_resource

#### 5.1.7.19.1 resource\_with\_material to material (as resource\_material)

AIM element: PATH  
Reference Path: resource\_with\_material<=  
generic\_manufacturing\_resource<=  
action\_resource <-  
requirement\_for\_action\_resource.resources [i]  
requirement\_for\_action\_resource <=  
action\_resource\_requirement  
action\_resource\_requirement.operation ->  
characterized\_action\_definition=action  
action =>  
property\_process <-  
process\_property\_association.process  
process\_property\_association  
process\_property\_association.description='resource material'  
process\_property\_association.property\_or\_shape ->  
property\_definition  
property\_definition.definition ->  
characterized\_definition  
characterized\_definition=characterized\_product\_definition  
characterized\_product\_definition=product\_definition  
product\_definition



### 5.1.7.20 Resource\_with\_representation

AIM element: resource\_with\_representation  
 Source: ISO 10303-223  
 Reference Path: resource\_with\_representation<=  
 generic\_manufacturing\_resource<=  
 action\_resource

#### 5.1.7.20.1 resource\_with\_representation\_to\_externally\_defined\_representation (as resource\_documentation)

AIM element: PATH  
 Reference Path: resource\_with\_representation<=  
 generic\_manufacturing\_resource<=  
 action\_resource  
 characterized\_resource\_definition=action\_resource  
 characterized\_resource\_definition <-  
 resource\_property.resource  
 resource\_property <-  
 resource\_property\_representation.property  
 resource\_property\_representation  
 resource\_property\_representation.representation ->  
 representation=>  
 externally\_defined\_representation\_with\_parameters

### 5.1.7.21 Shakeout

AIM element: shakeout  
 Source: ISO 10303-223  
 Reference Path: shakeout<=  
 casting\_equipment<=  
 machine<=  
 action\_resource

## 5.1.8 Manufacturing\_part\_properties UoF

### 5.1.8.1 Casting\_weight\_requirement

AIM element: property\_definition  
 Source: ISO 10303-41  
 Reference Path: {property\_definition  
 property\_definition.name = 'casting weight requirement'}

### 5.1.8.1.1 weight\_declared

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
{property\_definition\_representation=>  
material\_property\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i]->  
representation\_item=>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'actual')  
(descriptive\_representation\_item.description = 'estimated')}

### 5.1.8.1.2 casting\_weight\_requirement\_to\_property\_parameter (as property\_characteristics)

AIM element: PATH  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
{property\_definition\_representation=>  
material\_property\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i]->  
representation\_item  
(descriptive\_representation\_item)  
(measure\_representation\_item)

### 5.1.8.2 Composition\_element

AIM element: product\_definition  
 Source: ISO 10303-41  
 Reference Path: product\_definition

```
{product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='material composition definition'}
```

```
{product_definition
characterized_product_definition=product_definition
characterized_definition=characterized_product_definition<-
property_definition.definition
property_definition}
property_definition=>
{product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition <-
material_designation.definitions[i]}
```

#### 5.1.8.2.1 chemical\_element

AIM element: product.name  
 Source: ISO 10303-43  
 Reference Path: product\_definition  
 product\_definition.formation->  
 product\_definition\_formation  
 product\_definition\_formation.of\_product->  
 product  
 product.name

#### 5.1.8.2.2 compositon\_element to numeric\_parameter (as amount)

AIM element: PATH  
 Reference Path: product\_definition<-  
 product\_definition\_relationship.relating\_product\_definition  
 product\_definition\_relationship=>  
 product\_material\_composition\_relationship  
 product\_material\_composition\_relationship.constituent\_amount->  
 measure\_with\_unit

### 5.1.8.3 Data\_curve

AIM element: data\_curve  
Source: ISO 10303-223  
Reference Path: compound\_representation\_item=>  
data\_curve

#### 5.1.8.3.1 interpolation\_method

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: data\_curve<=  
compound\_representation\_item  
compound\_representation\_item.item\_element ->  
set\_representation\_item  
set\_representation\_item[i] ->  
representation\_item =>  
{representation\_item.name = 'interpolation method'}}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

#### 5.1.8.3.2 x\_unit

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: data\_curve<=  
compound\_representation\_item  
compound\_representation\_item.item\_element ->  
set\_representation\_item  
set\_representation\_item[i] ->  
representation\_item =>  
{representation\_item.name = 'x unit'}}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.8.3.3 y\_unit

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: data\_curve<=  
   compound\_representation\_item  
   compound\_representation\_item.item\_element ->  
   set\_representation\_item  
   set\_representation\_item[i] ->  
   representation\_item =>  
   {representation\_item.name = 'y unit'}}  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description

### 5.1.8.3.4 data\_curve to curve\_point (as data\_points)

AIM element: PATH  
 Reference Path: data\_curve<=  
   compound\_representation\_item  
   compound\_representation\_item.item\_element ->  
   set\_representation\_item  
   set\_representation\_item[i] ->  
   representation\_item =>  
   (representation\_item.name = 'data points')  
   geometric\_representation\_item=>  
   point=>  
   cartesian\_point

### 5.1.8.4 Descriptive\_parameter

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-45

#### 5.1.8.4.1 descriptive\_string

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45

### 5.1.8.5 Geometry\_characteristics

AIM element: product\_definition\_shape  
 Source: ISO 10303-41

### 5.1.8.5.1 geometry\_characteristics to direction\_element (as orientation)

AIM element: PATH  
Reference Path: product\_definition\_shape<=  
property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i]->  
representation\_item=>  
geometric\_representation\_item=>  
direction

### 5.1.8.5.2 geometry\_characteristics to numeric\_range(as characteristic\_size)

AIM element: PATH  
Reference Path: product\_definition\_shape<=  
property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i]->  
representation\_item  
measure\_representation\_item

### 5.1.8.5.3 geometry\_characteristics to shape\_element (as shape)

AIM element: PATH  
Reference Path: product\_definition\_shape <-  
shape\_aspect.of\_shape  
shape\_aspect

### 5.1.8.6 Hardness

AIM element: material\_property\_representation  
Source: SO 10303-45  
Reference path: material\_property\_representation  
material\_property\_representation.dependent\_environment->  
data\_environment}

### 5.1.8.6.1 high\_value

AIM element: measure\_representation\_item  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
 Reference path: material\_property\_representation<=  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'high value'}  
 {representation\_item  
 representation\_item=>  
 qualified\_representation\_item  
 qualified\_representation\_item.qualifiers[i]->  
 value\_qualifier=type\_qualifier  
 type\_qualifier  
 type\_qualifier.name='high value'}  
 representation\_item =>  
 measure\_representation\_item

### 5.1.8.6.2 low\_value

AIM element: measure\_representation\_item  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
 Reference path: material\_property\_representation<=  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'low value'}  
 {representation\_item  
 representation\_item=>  
 qualified\_representation\_item  
 qualified\_representation\_item.qualifiers[i]->  
 value\_qualifier=type\_qualifier  
 type\_qualifier  
 type\_qualifier.name='low value'}  
 representation\_item =>  
 measure\_representation\_item

### 5.1.8.6.3 nominal

AIM element:      measure\_representation\_item  
Source:            ISO 10303-45  
Rules:             dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
Reference path:    material\_property\_representation<=  
                    property\_definition\_representation  
                    property\_definition\_representation.used\_representation->  
                    representation  
                    representation.items[i] ->  
                    {representation\_item  
                    representation\_item.name = 'nominal'}  
                    {representation\_item  
                    representation\_item=>  
                    qualified\_representation\_item  
                    qualified\_representation\_item.qualifiers[i]->  
                    value\_qualifier=type\_qualifier  
                    type\_qualifier  
                    type\_qualifier.name='nominal'}  
                    representation\_item =>  
                    measure\_representation\_item

### 5.1.8.6.4 scale

AIM element:      data\_environment.description  
Source:            ISO 10303-45  
Reference path:    material\_property\_representation  
                    material\_property\_representation.dependent\_environment->  
                    data\_environment  
                    data\_environment.name = 'hardness'  
                    data\_environment.description



### 5.1.8.7 Material

AIM element: product\_definition  
 Source: ISO 10303-41  
 Reference Path: product\_definition  
   {product\_definition  
   product\_definition.frame\_of\_reference->  
   product\_definition\_context<=  
   application\_context\_element  
   application\_context\_element.name='material definition'}  
   {product\_definition  
   characterized\_product\_definition=product\_definition  
   characterized\_definition=characterized\_product\_definition<-  
   property\_definition.definition  
   property\_definition}  
   property\_definition=>  
   {product\_definition\_shape <=  
   property\_definition  
   property\_definition.definition ->  
   characterized\_definition <-  
   material\_designation.definitions[i]}

#### 5.1.8.7.1 material\_description

AIM element: product\_definition.description  
 Source: ISO 10303-41

#### 5.1.8.7.2 material\_id

AIM element: material\_designation.name  
 Source: ISO 10303-45  
 Reference Path: product\_definition  
   characterized\_product\_definition = product\_definition  
   characterized\_product\_definition  
   characterized\_definition = characterized\_product\_definition  
   characterized\_definition <-  
   material\_designation.definitions[i]  
   material\_designation  
   material\_designation.name

### 5.1.8.7.3 material to composition\_element(as composition)

AIM element: PATH  
Reference Path: product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition=>  
material\_property=>  
composition element

### 5.1.8.7.4 material to material\_property (as material\_characteristics)

AIM element: PATH  
Reference Path: product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition=>  
material\_property

### 5.1.8.7.5 material to specification (as material\_specification)

AIM element: PATH  
Reference Path: product\_definition =>  
product\_definition\_with\_associated\_documents  
product\_definition\_with\_associated\_documents.documentation\_ids[i] ->  
document

### 5.1.8.8 Material\_composition

AIM element: product\_material\_composition\_relationship  
Source: ISO 10303-45  
Reference path: product\_material\_composition\_relationship<=  
product\_definition\_relationship  
product\_definition\_relationship.id='material composition element'

**5.1.8.8.1 class**

AIM element: product\_material\_composition\_relationship.class  
 Source: ISO 10303-45  
 Reference path: product\_material\_composition\_relationship  
 product\_material\_composition\_relationship.class

**5.1.8.8.2 composition\_basis**

AIM element: product\_material\_composition\_relationship.composition\_bais  
 Source: ISO 10303-45  
 Reference path: product\_material\_composition\_relationship  
 product\_material\_composition\_relationship.composition\_bais

**5.1.8.8.3 description**

AIM element: product\_definition\_relationship.description  
 Source: ISO 10303-45  
 Reference path: product\_material\_composition\_relationship<=  
 product\_definition\_relationship  
 product\_definition\_relationship.description

**5.1.8.8.4 determination\_method**

AIM element: product\_material\_composition\_relationship.determination\_method  
 Source: ISO 10303-45  
 Reference path: product\_material\_composition\_relationship  
 product\_material\_composition\_relationship.determination\_method

**5.1.8.8.5 name**

AIM element: product\_definition\_relationship.name  
 Source: ISO 10303-45  
 Reference path: product\_material\_composition\_relationship<=  
 product\_definition\_relationship  
 product\_definition\_relationship.name

**5.1.8.8.6 material\_composition to composition\_element (as elements)**

AIM element: PATH  
 Reference path: product\_material\_composition\_relationship<=  
 product\_definition\_relationship  
 product\_definition\_relationship.relatng\_product\_definition->  
 product\_definition

### 5.1.8.9 Material\_property

AIM element: material\_property  
Source: ISO 10303-45

#### 5.1.8.9.1 alloy\_grade

AIM element: descriptive\_representation\_item  
Reference path: material\_property <=  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
{property\_definition\_representation=>  
material\_property\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i]->  
representation\_item=>  
representation\_item.name='alloy grade'  
descriptive\_representation\_item

#### 5.1.8.9.2 material\_property\_to\_hardness (as material\_hardness)

AIM element: PATH  
Reference path: material\_property <=  
property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation=>  
material\_property\_representation

### 5.1.8.9.3 material\_property to property\_parameter (as property\_-characteristics)

AIM element: PATH  
 Reference Path: material\_property <=  
 property\_definition  
 property\_definition = represented\_definition  
 represented\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation=>  
 material\_property\_representation}  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i]->  
 representation\_item=>  
 (descriptive\_representation\_item)  
 (measure\_representation\_item)

### 5.1.8.10 Material\_structure

AIM element: material\_structure  
 Source: ISO 10303-223  
 Reference Path: property\_definition=>  
 material\_property=>  
 material\_structure

#### 5.1.8.10.1 material\_structure to material\_structure\_element (as element)

AIM element: PATH  
 Reference Path: material\_structure=>  
 material\_property  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_product\_definition  
 characterized\_product\_definition  
 characterized\_product\_definition=product\_definition  
 product\_definition

### 5.1.8.10.2 material\_structure to illustration (as illustration)

AIM element: PATH  
eference Path: material\_structure=>  
material\_property  
property\_definition  
document\_reference\_item=property\_definition  
document\_reference\_item<-  
applied\_document\_reference.items[i]  
applied\_document\_reference  
{applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document=>  
illustration

### 5.1.8.10.3 material\_structure to specification (as referenced\_to)

AIM element: PATH  
eference Path: material\_structure=>  
material\_property  
property\_definition  
document\_reference\_item=property\_definition  
document\_reference\_item<-  
applied\_document\_reference.items[i]  
applied\_document\_reference  
{applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document

### 5.1.8.11 Material\_structure\_element

AIM element: product\_definition  
 Source: ISO 10303-41  
 Reference Path: product\_definition

```
{product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='material structure definition'}
```

```
{product_definition
characterized_product_definition=product_definition
characterized_definition=characterized_product_definition<-
property_definition.definition
property_definition}
property_definition=>
{product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition <-
material_designation.definitions[i]}
```

#### 5.1.8.11.1 type\_of\_element

AIM element: product\_definition.description  
 Source: ISO 10303-45  
 Reference Path: product\_definition  
 product\_definition.description

#### 5.1.8.11.2 material\_structure\_element to geometry\_characteristics (as characteristic\_geometry)

AIM element: PATH  
 Reference Path: physical\_structdure\_element<=  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape

### 5.1.8.11.3 material\_structure\_element to material (as made\_of)

AIM element: PATH  
Reference Path: product\_definition<-  
product\_definition\_relationship.related\_product\_definition  
product\_definition\_relationship  
(product\_definition\_relationship.id='material structure element')  
{product\_definition\_relationship=>  
product\_material\_composition\_relationship}  
product\_definition\_relationship.relying\_product\_definition->  
product\_definition

### 5.1.8.11.4 material\_structure\_element to numeric\_range (as amount)

AIM element: PATH  
Reference Path: product\_definition<-  
product\_definition\_relationship.related\_product\_definition  
product\_definition\_relationship  
(product\_definition\_relationship.id='material structure element')  
product\_definition\_relationship=>  
product\_material\_composition\_relationship  
product\_material\_composition\_relationship.constituent\_amount->  
measure\_with\_unit=>  
measure\_representation\_item

### 5.1.8.12 Numeric\_parameter

AIM element: measure\_representation\_item  
Source: ISO 10303-45

#### 5.1.8.12.1 parameter\_description

AIM element: (measure\_qualification.description)  
Source: ISO 10303-41  
Reference Path: measure\_representation\_item<=  
measure\_with\_unit<-  
measure\_qualification.qualified\_measure  
measure\_qualification  
measure\_qualification.name='numeric parameter clarification'  
measure\_qualification.description



### 5.1.8.12.2 parameter\_note

AIM element: (type\_qualifier.name)  
 Source: ISO 10303-41  
 Reference Path: measure\_representation\_item <=  
                   measure\_with\_unit <=  
                   measure\_qualification.qualified\_measure  
                   measure\_qualification  
                   measure\_qualification.name='numeric parameter clarification'  
                   measure\_qualification.qualifiers[i]->  
                   value\_qualifier=type\_qualifier  
                   type\_qualifier  
                   type\_qualifier.name

### 5.1.8.12.3 parameter\_unit

AIM element: named\_unit  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_named\_unit – (See 5.2.4.9)  
 Reference Path: measure\_representation\_item <=  
                   measure\_with\_unit  
                   measure\_with\_unit.unit\_component ->  
                   unit  
                   unit = named\_unit  
                   named\_unit

### 5.1.8.12.4 parameter\_value

AIM element: measure\_value  
 Source: ISO 10303-41  
 Reference Path: measure\_representation\_item <=  
                   measure\_with\_unit  
                   measure\_with\_unit.value\_component ->  
                   measure\_value

### 5.1.8.12.5 significant\_digit

AIM element: (precision\_qualifier.precision\_valu)  
Source: ISO 10303-41  
Reference Path: measure\_representation\_item<=  
measure\_with\_unit<-  
measure\_qualification.qualified\_measure  
measure\_qualification  
measure\_qualification.name='numeric parameter clarification'  
measure\_qualification.qualifiers[i]->  
value\_qualifier=precision\_qualifier  
precision\_qualifier  
precision\_qualifier.precision\_value

### 5.1.8.13 Numeric\_parameter\_with\_tolerance

AIM element: [measure\_representation\_item]  
[qualified\_representation\_item]  
Source: ISO 10303-45  
ISO 10303-45

#### 5.1.8.13.1 numeric\_parameter\_with\_tolerance to plus\_minus\_value (as implicit\_tolerance)

AIM element: IDENTICAL MAPPING

#### 5.1.8.13.2 numeric\_parameter\_with\_tolerance to tolerance\_limit (as implicit\_tolerance)

AIM element: PATH  
Rules: dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
Reference Path: qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->  
value\_qualifier  
value\_qualifier = type\_qualifier  
type\_qualifier

#### 5.1.8.13.3 numeric\_parameter\_with\_tolerance to limits\_and\_fits (as implicit\_tolerance)

AIM element: IDENTICAL MAPPING

#### 5.1.8.13.4 numeric\_parameter\_with\_tolerance to tolerance\_range (as implicit\_tolerance)

AIM element: IDENTICAL MAPPING

#### 5.1.8.14 Numeric\_range

AIM element: measure\_representation\_item  
Source: ISO 10303-45

##### 5.1.8.14.1 numeric\_range to tolerance\_range (as value\_range)

AIM element: IDENTICAL MAPPING

#### 5.1.8.15 Part\_property

AIM element: property\_definition  
Source: ISO 10303-41  
Reference Path: {property\_definition  
property\_definition.name = 'part property'}

##### 5.1.8.15.1 part\_property to property\_parameter (as process\_characteristics)

AIM element: PATH  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
(descriptive\_representation\_item)  
(measure\_representation\_item)

#### 5.1.8.16 Process\_property

AIM element: property\_definition  
Source: ISO 10303-41  
Reference Path: {property\_definition  
property\_definition.name = 'process property'}

### 5.1.8.16.1 process\_name

AIM element: representation.name  
Source: ISO 10303-43  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.name

### 5.1.8.16.2 process\_property to property\_parameter (as process\_characteristics)

AIM element: PATH  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
(descriptive\_representation\_item)  
(measure\_representation\_item)

## 5.1.8.17 Property

AIM element: property\_definition  
Source: ISO 10303-41

### 5.1.8.17.1 description

AIM element: property\_definition.name  
Source: ISO 10303-41

### 5.1.8.17.2 property\_name

AIM element: property\_definition.name  
Source: ISO 10303-41

### 5.1.8.17.3 property to casting\_weight\_requirement (as weight\_characteristics)

AIM element: PATH  
 Reference Path: property\_definition <-  
   property\_definition\_relationship.relating\_property\_definition  
   property\_definition\_relationship  
   property\_definition\_relationship.related\_property\_definition ->  
   property\_definition  
   {property\_definition  
   property\_definition.name = 'casting weight requirement'}

### 5.1.8.17.4 property to material\_property (as material\_characteristics)

AIM element: PATH  
 Reference Path: property\_definition <-  
   property\_definition\_relationship.relating\_property\_definition  
   property\_definition\_relationship  
   property\_definition\_relationship.related\_property\_definition ->  
   property\_definition =>  
   material\_property

### 5.1.8.17.5 property to part\_property (as part\_characteristics)

AIM element: PATH  
 Reference Path: property\_definition <-  
   property\_definition\_relationship.relating\_property\_definition  
   property\_definition\_relationship  
   property\_definition\_relationship.related\_property\_definition ->  
   property\_definition  
   {property\_definition  
   property\_definition.name = 'part property'}

### 5.1.8.17.6 property to process\_property (as process\_characteristics)

AIM element: PATH  
 Reference Path: property\_definition <-  
   property\_definition\_relationship.relating\_property\_definition  
   property\_definition\_relationship  
   property\_definition\_relationship.related\_property\_definition ->  
   property\_definition  
   {property\_definition  
   property\_definition.name = 'process property'}

### 5.1.8.17.7 property to surface\_property (as surface\_characteristics)

AIM element: PATH  
Reference Path: property\_definition <-  
property\_definition\_relationship.relatng\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition ->  
property\_definition  
{property\_definition  
property\_definition.name = 'surface property'}

### 5.1.8.17.8 property to shape\_aspect (as shape\_property\_characteristics)

AIM element: PATH  
Reference Path: property\_definition  
property\_definition.definition ->  
characterized\_definition  
characterized\_definition = shape\_definition  
shape\_definition  
shape\_definition = shape\_aspect  
shape\_aspect

### 5.1.8.17.9 property to specification (as property\_description)

AIM element: PATH  
Reference Path: property\_definition  
document\_reference\_item = property\_definition  
document\_reference\_item <-  
applied\_document\_reference.items[i]  
applied\_document\_reference <=  
document\_reference  
document\_reference.assigned\_document ->  
document=>  
process\_plan\_specification

### 5.1.8.18 Property\_relationship

AIM element: property\_relationship  
Source: ISO 10303-223  
Reference Path: property\_relationship=>  
representation

**5.1.8.18.1 x\_property\_name**

AIM element:      measure\_representation\_item  
 Source:            ISO 10303-45  
 Reference Path:    property\_relationship=>  
                     representation  
                     representation.items[i] ->  
                     representation\_item  
                     {representation\_item  
                     representation\_item.name = 'x property name'}  
                     (measure\_representation\_item)

**5.1.8.18.2 y\_property\_name**

AIM element:      measure\_representation\_item  
 Source:            ISO 10303-45  
 Reference Path:    property\_relationship=>  
                     representation  
                     representation.items[i] ->  
                     representation\_item  
                     {representation\_item  
                     representation\_item.name = 'y property name'}  
                     (measure\_representation\_item)

**5.1.8.18.3 property\_relationship to data\_curve(as relationship\_curve)**

AIM element:      PATH  
 Reference Path:    property\_relationship=>  
                     representation  
                     representation.items[i]-.  
                     representation\_item  
                     compound\_representation\_item  
                     data\_curve

**5.1.8.19 Property\_parameter**

AIM element:      (descriptive\_representation\_item)  
                     (measure\_representation\_item)  
 Source:            ISO 10303-45  
                     ISO 10303-45  
 Reference Path:    (descriptive\_representation\_item <=)  
                     (measure\_representation\_item <=)  
                     representation\_item

### 5.1.8.19.1 parameter\_name

AIM element: representation\_item.name  
Source: ISO 10303-43

### 5.1.8.20 Surface\_property

AIM element: property\_definition  
Source: ISO 10303-41  
Reference Path: {property\_definition  
property\_definition.name = 'surface property'}

#### 5.1.8.20.1 casting\_appearance

AIM element: PATH  
Reference Path: property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
descriptive\_representation\_item  
(descriptive\_representation\_item.description = 'critical')  
(descriptive\_representation\_item.description = 'important')  
(descriptive\_representation\_item.description = 'not important')

#### 5.1.8.20.2 surface\_finish

AIM element: property\_definition  
Source: ISO 10303-41  
Reference Path: {property\_definition  
property\_definition.description = 'surface finish'}



### 5.1.8.20.3 surface\_property to property\_parameter (as process\_-characteristics)

AIM element: PATH  
 Reference Path: property\_definition  
                   property\_definition = represented\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   representation  
                   representation.items[i] ->  
                   representation\_item  
                   (descriptive\_representation\_item)  
                   (measure\_representation\_item)

## 5.1.9 Manufacturing\_process\_requirement\_documents UoF

### 5.1.9.1 Customer\_design\_deficiency\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
                   (document.kind->  
                   document\_type  
                   document\_type.product\_data\_type='customer design deficiency report')

### 5.1.9.1.1 list\_of\_results

AIM element: PATH  
Source: ISO 10303-41  
Reference Path: document=>  
document\_file <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition  
{property\_definition.name = 'document property'}  
represented\_definition = property\_definition  
represented\_definition <=  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
{representation.name = 'design defect report'}  
representation.items[i]->  
representation\_item=>  
{representation\_item.name='list of results'}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.9.2 Design\_reference

AIM element: design\_reference  
Source: ISO 10303-223  
Reference Path: design\_reference<=  
document

#### 5.1.9.2.1 drawing\_type

AIM element: document\_type.product\_data\_type  
Source: ISO 10303-41  
Reference Path: design\_reference<=  
document  
document.kind->  
document\_type  
document\_type.product\_data\_type

**5.1.9.2.2 id**

AIM element: document.id  
 Source: ISO 10303-41  
 Reference Path: design\_reference<=  
                   document  
                   document.id

**5.1.9.2.3 revision\_level**

AIM element: document\_usage\_constraint.subject\_element\_value  
 Source: ISO 10303-41  
 Reference Path: design\_reference<=  
                   document <-  
                   document\_usage\_constraint.source  
                   document\_usage\_constraint  
                   {document\_usage\_constraint.subject\_element='drawing revision level'}  
                   document\_usage\_constraint.subject\_element\_value

**5.1.9.2.4 design\_reference to document\_file (as drawing\_data\_file)**

AIM element: IDENTICAL MAPPING

**5.1.9.2.5 design\_reference to activity (as identifies)**

AIM element: PATH  
 Reference Path: design\_reference<=  
                   document <-  
                   document\_reference.assigned\_document  
                   document\_reference =>  
                   applied\_document\_reference  
                   applied\_document\_reference.items [i]->  
                   document\_reference\_item= process\_plan\_activity  
                   process\_plan\_activity

### 5.1.9.2.6 design\_reference to part\_version (as documents\_version)

AIM element: PATH  
Reference Path: design\_reference<=  
document <-  
document\_product\_association.relatiing\_document  
document\_product\_association  
document\_product\_association.related\_product->  
product\_or\_formation\_or\_definition  
product\_or\_formation\_or\_definition=product\_definition  
product\_definition

### 5.1.9.3 Digital\_file

AIM element: document\_file  
Source: ISO 10303-223  
Reference Path: document\_file =>  
[document<-  
{document\_representation\_type.represented\_document  
document\_representation\_type  
document\_representation\_type.name = 'digital'}]  
[characterized\_object]

### 5.1.9.4 Document\_assignment

AIM element: applied\_document\_reference  
Source: ISO 10303-223  
Reference Path: applied\_document\_reference<=  
document\_reference

#### 5.1.9.4.1 role

AIM element: object\_role.name  
Source: ISO 10303-41  
Reference Path: applied\_document\_reference <=  
document\_reference  
role\_select=document\_reference  
role\_select<-  
role\_association.item\_with\_role  
role\_association  
role\_association.role->  
object\_role  
object\_role.name

### 5.1.9.4.2 document\_assignment to document\_file (as assigned\_document)

AIM element: PATH  
 Reference Path: applied\_document\_reference<=  
                   document\_reference  
                   document\_reference.assigned\_document->  
                   document=>  
                   document\_file

### 5.1.9.5 Document\_file

AIM element: document\_file  
 Source: ISO 10303-223  
 Rules: document\_file\_rules – (See 5.2.4.15)  
 Reference Path: document\_file =>  
                   [document<-  
                   {document\_representation\_type.represented\_document  
                   document\_representation\_type  
                   (document\_representation\_type.name = 'digital')  
                   (document\_representation\_type.name = 'physical')}]  
                   [characterized\_object]

#### 5.1.9.5.1 document\_type

AIM element: document\_type.product\_data\_type  
 Source: ISO 10303-41  
 Reference Path: document\_file<=  
                   document  
                   document.kind->  
                   document\_type  
                   document\_type.product\_data\_type

#### 5.1.9.5.2 file\_id

AIM element: document.id  
 Source: ISO 10303-41  
 Reference Path: document\_file<=  
                   document  
                   document.id

### 5.1.9.5.3 version\_id

AIM element: identification\_assignment.assigned\_id  
Source: ISO 10303-41  
Reference Path: document\_file  
identification\_assignment\_item = document\_file  
identification\_assignment\_item <-  
applied\_identification\_assignment.items[i]  
applied\_identification\_assignment <=  
identification\_assignment  
{identification\_assignment.role ->  
identification\_role  
identification\_role.name = 'version'}  
identification\_assignment.assigned\_id

### 5.1.9.5.4 document\_file to document\_file\_properties (as document\_properties)

AIM element: PATH  
Source: ISO 10303-41  
Reference Path: document\_file <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <-  
property\_definition.definition  
property\_definition  
{property\_definition.name = 'document property'}  
represented\_definition = property\_definition  
represented\_definition <-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
{representation.name = 'document properties'}

### 5.1.9.5.5 document\_file to external\_file\_identification (as file\_location)

AIM element: PATH  
 Source: ISO 10303-41  
 Reference Path: document\_file<=  
                   document  
                   external\_identification\_item=document  
                   external\_identification\_item<-  
                   applied\_external\_identification\_assignment.items[i]  
                   applied\_external\_identification\_assignment<=  
                   external\_identification\_assignment  
                   {external\_identification\_assignment<=  
                   identification\_assignment}

### 5.1.9.6 Document\_file\_properties

AIM element: representation  
 Source: ISO 10303-43

#### 5.1.9.6.1 content\_country\_code

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: representation  
                   {representation.name='document content'}  
                   representation.items[i]->  
                   representation\_item=>  
                   {representation\_item.name='country code'}  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

#### 5.1.9.6.2 content\_detail\_level

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: representation  
                   {representation.name='document content'}  
                   representation.items[i]->  
                   representation\_item=>  
                   {representation\_item.name='detail level'}  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

### 5.1.9.6.3 content\_geometry\_type

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i]->  
representation\_item=>  
{representation\_item.name='geometry type'}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.9.6.4 content\_language\_code

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i]->  
representation\_item=>  
{representation\_item.name='language code'}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.9.6.5 creating\_interface

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i]->  
representation\_item=>  
{representation\_item.name='creating interface'}  
descriptive\_representation\_item  
descriptive\_representation\_item.description



**5.1.9.6.6 creating\_operating\_system**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: representation  
                   {representation.name='document content'}  
                   representation.items[i]->  
                   representation\_item=>  
                   {representation\_item.name='creating operating system'}  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

**5.1.9.6.7 creating\_system**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: representation  
                   {representation.name='document content'}  
                   representation.items[i]->  
                   representation\_item=>  
                   {representation\_item.name='creating system'}  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

**5.1.9.6.8 data\_format**

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-45  
 Reference Path: representation  
                   {representation.name='document content'}  
                   representation.items[i]->  
                   representation\_item=>  
                   {representation\_item.name='data format'}  
                   descriptive\_representation\_item  
                   descriptive\_representation\_item.description

### 5.1.9.6.9 file\_size

AIM element: measure\_representation\_item  
Source: ISO 10303-41  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i] ->  
representation\_item =>  
{representation\_item  
representation\_item.name = 'file size'}  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
ratio\_measure\_with\_unit}

### 5.1.9.6.10 format\_character\_code

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-45  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i]->  
representation\_item=>  
{representation\_item.name='format character code'}  
descriptive\_representation\_item  
descriptive\_representation\_item.description

### 5.1.9.6.11 page\_count

AIM element: count\_measure  
Source: ISO 10303-41  
Reference Path: representation  
{representation.name='document content'}  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'page count'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.value\_component ->  
measure\_value  
measure\_value = count\_measure  
count\_measure}

### 5.1.9.7 Document\_file\_relationship

AIM element: document\_relationship  
Source: ISO 10303-41

#### 5.1.9.7.1 previous\_file

AIM element: document\_file  
Source: ISO 10303-41  
Reference Path: document\_relationship  
document\_relationship.related\_document->  
document=>  
document\_file

#### 5.1.9.7.2 succeeding\_file

AIM element: document\_file  
Source: ISO 10303-41  
Reference Path: document\_relationship  
document\_representation.relatng\_document->  
document=>  
document\_file

### 5.1.9.8 Executable

AIM element: externally\_defined\_schema  
Source: ISO 10303-223  
Reference path: externally\_defined\_schema<=  
externally\_defined\_item  
externally\_defined\_item.item\_id='executable'

#### 5.1.9.8.1 executable\_id

AIM element: external\_source.source\_id  
Source: ISO 10303-41  
Reference path: externally\_defined\_schema<=  
externally\_defined\_item  
externally\_defined\_item.source->  
external\_source  
external\_source.description='executable id'  
external\_source.source\_id

### 5.1.9.9 External\_file\_identification

AIM element: applied\_external\_identification\_assignment  
Source: ISO 10303-223  
Reference Path: applied\_external\_identification\_assignment<=  
external\_identification\_assignment<=  
identification\_assignment

#### 5.1.9.9.1 external\_file\_id

AIM element: identification\_assignment.assigned\_id  
Source: ISO 10303-41  
Reference Path: applied\_external\_identification\_assignment<=  
external\_identification\_assignment<=  
identification\_assignment  
identification\_assignment.assigned\_id

#### 5.1.9.9.2 external\_file\_location

AIM element: external\_source.source\_id  
Source: ISO 10303-41  
Reference Path: applied\_external\_identification\_assignment <=  
external\_identification\_assignment  
{external\_identification\_assignment <=  
identification\_assignment}  
external\_identification\_assignment.source ->  
external\_source  
external\_source.source\_id

### 5.1.9.10 External\_schema\_definition

AIM element: externally\_defined\_schema  
Source: ISO 10303-223  
Rules: restrict\_name\_for\_known\_source – (See 5.2.4.29)  
Reference Path: [externally\_defined\_schema<=  
externally\_defined\_item  
{externally\_defined\_item.source ->  
external\_source =>  
known\_source <=  
pre\_defined\_item  
pre\_defined\_item.name = 'ISO 10303 part'}]

### 5.1.9.10.1 external\_schema\_definition to document\_assignment (as reference\_schema)

AIM element: PATH  
 Reference Path: externally\_defined\_schema  
 externally\_defined\_schema = document\_reference\_item  
 document\_reference\_item<-  
 applied\_document\_reference.items[i]  
 applied\_document\_reference  
 {applied\_document\_reference<=  
 document\_reference  
 document\_reference.assigned\_document->  
 document  
 document.description = 'externally defined schema' }

### 5.1.9.11 Product\_quality\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
 (document.kind->  
 document\_type  
 document\_type.product\_data\_type='product quality report')  
 {document<-  
 document\_reference.assigned\_document  
 document\_reference=>  
 applied\_document\_reference}

### 5.1.9.11.1 action\_required

AIM element: measure\_representation\_item  
Source: ISO 10303-41  
Reference Path: product\_quality\_report<=  
document=>  
document\_file<=  
characterized\_object  
characterized\_definition=characterized\_object  
characterized\_definition<=  
property\_definition.definition  
property\_definition  
represented\_definition=property\_definition<=  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.representation  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'action required'}  
representation\_item =>  
descriptive\_representation\_item

### 5.1.9.11.2 description

AIM element: document.description  
Source: ISO 10303-41  
Reference Path: document  
document.description

### 5.1.9.11.3 number\_failed

AIM element: measure\_representation\_item  
 Source: ISO 10303-41  
 Reference Path: document=>  
 document\_file<=  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition<=  
 property\_definition.definition  
 property\_definition  
 represented\_definition=property\_definition<=  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.representation  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'number failed'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure}

### 5.1.9.11.4 Product\_quality\_report to inspection\_or\_test\_result (as what\_failed)

AIM element: PATH  
 Reference Path: document=>  
 document\_file<=  
 characterized\_object  
 characterized\_definition=characterized\_object  
 characterized\_definition<=  
 property\_definition.definition  
 property\_definition=>  
 inspection\_or\_test\_result

### 5.1.9.12 Hardcopy

AIM element: document\_file  
Source: ISO 10303-223  
Reference Path: document\_file =>  
[document<-  
{document\_representation\_type.represented\_documentmnt  
document\_representation\_type  
document\_representation\_type.name = 'physical'}]  
[characterized\_object]

### 5.1.9.13 Illustration

AIM element: illustration  
Source: ISO 10303-223  
Reference Path: illustration<=  
document

#### 5.1.9.13.1 description

AIM element: document.description  
Source: ISO 10303-41  
Reference Path: illustration<=  
document  
document.description

#### 5.1.9.13.2 id

AIM element: document.id  
Source: ISO 10303-41  
Reference Path: illustration<=  
document  
document.id



**5.1.9.13.3 illustration to view\_reference (as is\_owned\_by)**

AIM element: PATH  
 Source: ISO 10303-41  
 Reference Path: illustration  
 document\_usage\_constraint\_item=illustration  
 document\_usage\_constraint\_item<-  
 applied\_document\_usage\_constraint\_assignment.items[i]  
 applied\_document\_usage\_constraint\_assignment  
 applied\_document\_usage\_constraint\_assignment<=  
 document\_usage\_constraint\_assignment  
 document\_usage\_constraint\_assignment.assigned\_document\_usage->  
 document\_usage\_constraint=>  
 view\_reference

**5.1.9.14 Part\_dimensioning\_standard**

AIM element: applied\_document\_reference  
 Source: ISO 10303-223  
 Reference Path: applied\_document\_reference<=  
 document\_reference  
 {document\_reference.role->  
 object\_role  
 object\_role.name= 'dimensioning standard'}

**5.1.9.14.1 Part\_dimensioning\_standard to Part\_version (as applied\_part)**

AIM element: PATH  
 Source: ISO 10303-41  
 Reference Path: applied\_document\_reference  
 applied\_document\_reference.items [i]->  
 document\_reference\_item  
 document\_reference\_item=product\_definition\_formation  
 product\_definition\_formation

### 5.1.9.15 Proof\_report

AIM element: document  
Source: ISO 10303-41  
Reference Path: document  
(document.kind->  
document\_type  
document\_type.product\_data\_type='proof report')  
{document<-  
document\_reference.assigned\_document  
document\_reference=>  
applied\_document\_reference}

### 5.1.9.16 Quality\_acceptance\_report

AIM element: document  
Source: ISO 10303-41  
Reference Path: document  
(document.kind->  
document\_type  
document\_type.product\_data\_type='quality acceptance report')  
{document<-  
document\_reference.assigned\_document  
document\_reference=>  
applied\_document\_reference}

#### 5.1.9.16.1 quality\_acceptance\_report to reporting\_requirement (as applied - to)

AIM element: PATH  
Source: ISO 10303-41  
Reference Path: document<-  
document\_reference.assigned\_document  
document\_reference=>  
applied\_document\_reference  
applied\_document\_reference.items[i]->  
document\_reference\_item  
document\_reference\_item=reporting\_requirement  
reporting\_requirement

### 5.1.9.17 Simulation\_exception\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
 (document.kind->  
 document\_type  
 document\_type.product\_data\_type='simulation acceptance report')  
 {document<-  
 document\_reference.assigned\_document  
 document\_reference=>  
 applied\_document\_reference}

### 5.1.9.18 Simulation\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
 (document.kind->  
 document\_type  
 document\_type.product\_data\_type='simulation report')  
 {document<-  
 document\_reference.assigned\_document  
 document\_reference=>  
 applied\_document\_reference}

### 5.1.9.19 Special\_instruction

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-45  
 Reference Path: descriptive\_representation\_item<=  
 representation\_item  
 representation\_item.name='special instruction'

#### 5.1.9.19.1 instruction\_text

AIM element: descriptive\_representation\_item.description  
 Source: ISO 10303-41

### 5.1.9.19.2 instruction\_type

AIM element: description\_attribute.attribute\_value  
Source: ISO 10303-41  
Reference Path: descriptive\_representation\_item<=  
representation\_item<-  
representation.items[i]  
representation  
description\_attribute\_select=representation  
description\_attribute\_select<-  
description\_attribute.described\_item  
description\_attribute  
description\_attribute.attribute\_value

### 5.1.9.20 Specification

AIM element: document  
Source: ISO 10303-41

#### 5.1.9.20.1 specification\_class

AIM element: document\_with\_class.class  
Source: ISO 10303-41  
Reference Path: document=>  
document\_with\_class

#### 5.1.9.20.2 specification\_description

AIM element: document.description  
Source: ISO 10303-41  
Reference Path: document  
document.description

#### 5.1.9.20.3 specification\_id

AIM element: document.id  
Source: ISO 10303-41  
Reference Path: document  
document.id

**5.1.9.20.4 version\_id**

AIM element: identification\_assignment.assigned\_id  
 Source: ISO 10303-41  
 Reference Path: document  
 identification\_item = document  
 identification\_item <-  
 applied\_identification\_assignment.items[i]  
 applied\_identification\_assignment <=  
 identification\_assignment  
 {identification\_assignment.role ->  
 identification\_role  
 identification\_role.name = 'version'}  
 identification\_assignment.assigned\_id

**5.1.9.20.5 specification to date (as release\_date)**

AIM element: date  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: document  
 date\_item = document  
 date\_item <-  
 applied\_date\_assignment.items [i]  
 applied\_date\_assignment <=  
 date\_assignment  
 {date\_assignment.role ->  
 date\_role  
 date\_role.name='classification date'}  
 date\_assignment.assigned\_date ->  
 date

**5.1.9.20.6 specification to document\_assignment (as specification\_document)**

AIM element: PATH  
 Reference Path: process\_plan\_specification<=  
 document<-  
 document\_reference.assigned\_document  
 document\_reference=>  
 applied\_document\_reference

### **5.1.9.20.7 specification to organization (as issuing\_organization)**

AIM element: PATH  
Reference Path: document  
organization\_item=document  
organization\_item <-  
applied\_organization\_assignment.items [i]<=  
applied\_organization\_assignment <=  
organization\_assignment  
organization\_assignment.assigned\_organization->  
organization

### **5.1.9.20.8 specification to specification\_usage\_constraint (as constraint)**

AIM element: PATH  
Reference Path: document <-  
document\_usage\_constraint.source  
document\_usage\_constraint

### **5.1.9.21 Specification\_usage\_constraint**

AIM element: document\_usage\_constraint  
Source: ISO 10303-41

#### **5.1.9.21.1 class\_id**

AIM element: document\_usage\_constraint.subject\_element\_value  
Source: ISO 10303-41

#### **5.1.9.21.2 element**

AIM element: document\_usage\_constraint.subject\_element  
Source: ISO 10303-41

### **5.1.9.22 Supplemental\_document**

AIM element: process\_plan\_specification  
Source: ISO 10303-223  
Reference Path: process\_plan\_specification<=  
document  
document.kind->  
document\_type  
document\_type.product\_data\_type='supplemental document'

### 5.1.9.23 Customer\_tool\_design\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
 (document.kind->  
 document\_type  
 document\_type.product\_data\_type='customer tool design report')

#### 5.1.9.23.1 customer\_tool\_design\_report to tooling (as designed\_tool)

AIM element: PATH  
 Source: ISO 10303-41  
 Reference Path: document <-  
 document\_product\_association.relatating\_document  
 document\_product\_association  
 document\_product\_association.related\_product->  
 product\_or\_formation\_or\_definition  
 product\_or\_formation\_or\_definition=product\_definition\_formation  
 product\_definition\_formation

### 5.1.9.24 Simulation\_tooling\_report

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: document  
 (document.kind->  
 document\_type  
 document\_type.product\_data\_type='simulation tooling report')  
 {document<-  
 document\_reference.assigned\_document  
 document\_reference=>  
 applied\_document\_reference}

### 5.1.9.25 View\_reference

AIM element: view\_reference  
 Source: ISO 10303-223  
 Reference Path: view\_reference<=  
 document\_usage\_constraint

### 5.1.9.25.1 sheet

AIM element: document\_usage\_constraint.subject\_element\_value  
Source: ISO 10303-41  
Reference Path: view\_reference<=  
document\_usage\_constraint  
{document\_usage\_constraint.subject\_element='sheet'}  
document\_usage\_constraint.subject\_element\_value

### 5.1.9.25.2 view

AIM element: document\_usage\_constraint.subject\_element\_value  
Source: ISO 10303-41  
Reference Path: view\_reference<=  
document\_usage\_constraint  
{document\_usage\_constraint.subject\_element='view'}  
document\_usage\_constraint.subject\_element\_value

### 5.1.9.25.3 zone

AIM element: document\_usage\_constraint.subject\_element\_value  
Source: ISO 10303-41  
Reference Path: view\_reference<=  
document\_usage\_constraint  
{document\_usage\_constraint.subject\_element='zone'}  
document\_usage\_constraint.subject\_element\_value

### 5.1.9.25.4 view\_reference to design\_reference(as owned\_by)

AIM element: PATH  
Source: ISO 10303-41  
Reference Path: view\_reference<=  
document\_usage\_constraint  
document\_usage\_constraint.source->  
document=>  
design\_reference



## 5.1.10 Measurement\_limitations UoF

### 5.1.10.1 Angular\_dimension\_tolerance

#1: if directed = true

AIM element: (directed\_dimensional\_location)  
(angular\_location)  
Source: ISO 10303-223  
Reference Path: (directed\_dimensional\_location<= )  
(angular\_location <= )  
dimensional\_location

#2: if directed = false

AIM element: (angular\_location)  
Source: ISO 10303-47  
Rules:

### 5.1.10.1.1 angular\_dimension\_tolerance to orientation (as plane\_and\_direction)

#1: if directed = true

AIM element: PATH  
Reference Path: (directed\_dimensional\_location <=  
dimensional\_location =>  
angular\_location)

#2: if directed = false

AIM element: PATH  
Reference Path: (angular\_location <= )  
dimensional\_location <=  
shape\_aspect\_relationship  
shape\_definition = shape\_aspect\_relationship  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'orientation'}  
representation\_item =>  
geometric\_representation\_item =>  
placement

### 5.1.10.2 Angular\_size\_dimension\_tolerance

AIM element: angular\_size  
Source: ISO 10303-47

**5.1.10.2.1 full\_or\_half**

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-43  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: angular\_size <=  
 dimensional\_size  
 dimensional\_characteristic = dimensional\_size  
 dimensional\_characteristic <=  
 dimensional\_characteristic\_representation.dimension  
 dimensional\_characteristic\_representation  
 dimensional\_characteristic\_representation.representation ->  
 shape\_dimension\_representation <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 representation\_item=>  
 descriptive\_representation\_item  
 descriptive\_representation\_item.description  
 {( descriptive\_representation\_item = 'full angle')  
 (descriptive\_representation\_item = 'half angle')}

**5.1.10.2.2 major\_angle**

AIM element: angular\_size.angle\_selection  
 Source: ISO 10303-47

**5.1.10.3 Common\_datum**

AIM element: {common\_datum<=  
 [datum]  
 [composite\_shape\_aspect]}  
 Source: ISO 10303-519

**5.1.10.3.1 common\_datum to datum\_feature (as element)**

AIM element: PATH  
 Reference Path: common\_datum  
 composite\_shape\_aspect  
 composite\_shape\_aspect.component\_relationships [i] ->  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.related\_shape\_aspect ->  
 shape\_aspect =>  
 datum

#### 5.1.10.4 Curved\_dimension\_tolerance

AIM element: dimensional\_size  
Source: ISO 10303-47  
Reference Path: {dimensional\_size  
dimensional\_size.name = 'curve length'}

#### 5.1.10.5 Datum

AIM element: datum\_reference  
Source: ISO 10303-47

##### 5.1.10.5.1 name

AIM element: shape\_aspect.name  
Source: ISO 10303-41  
Reference Path: datum\_reference  
datum\_reference.referenced\_datum ->  
datum <=  
shape\_aspect  
shape\_aspect.name

##### 5.1.10.5.2 precedence

AIM element: datum\_reference.precedence  
Source: ISO 10303-47

#### 5.1.10.6 Datum\_feature

AIM element: (datum)  
([datum]  
[datum\_feature])  
Source: ISO 10303-47  
Reference Path: {(datum)  
(datum <=  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relying\_shape\_aspect ->  
shape\_aspect =>  
datum\_feature)}

**5.1.10.6.1 datum\_feature to datum\_target\_set (as datum\_representation)**

AIM element: PATH  
 Reference Path: datum  
 datum.established\_by\_relationships

**5.1.10.6.2 datum\_feature to material\_condition\_modifier (as modifier)**

AIM element: PATH  
 Reference Path: datum <-  
 datum\_reference.referenced\_datum  
 datum\_reference =>  
 referenced\_modified\_datum  
 referenced\_modified\_datum.modifier ->  
 limit\_condition

**5.1.10.6.3 datum\_feature to shape\_element (as datum\_representation)**

AIM element: IDENTICAL MAPPING

**5.1.10.7 Datum\_target**

AIM element: datum\_target  
 Source: ISO 10303-47

**5.1.10.7.1 identifier**

AIM element: datum\_target.target\_id  
 Source: ISO 10303-47

**5.1.10.8 Datum\_target\_set**

AIM element: datum\_established\_by\_relationships  
 Source: ISO 10303-47  
 Reference Path: {datum.established\_by\_relationships[i] ->  
 shape\_aspect\_relationship  
 shape\_aspect\_relationship.relating\_shape\_aspect ->  
 shape\_aspect =>  
 datum\_target}

ISO 10303-223:2008 (E)

### 5.1.10.8.1 rule\_description

AIM element: shape\_aspect\_relationship.description  
Source: ISO 10303-41  
Reference Path: datum.established\_by\_relationships[i] ->  
shape\_aspect\_relationship  
shape\_aspect\_relationship.description

### 5.1.10.8.2 datum\_target\_set to datum\_target (as target\_shape)

AIM element: PATH  
Reference Path: datum.established\_by\_relationships  
datum <=  
shape\_aspect<-  
shape\_aspect\_relationship.related\_shape\_aspect  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
datum\_target

### 5.1.10.9 Diameter\_dimension\_tolerance

AIM element: dimensional\_size  
Source: ISO 10303-47  
Reference Path: {dimensional\_size  
dimensional\_size.name = 'diameter'}

### 5.1.10.10 Dimensional\_tolerance

#1: Dimensional\_tolerance as Dimensional\_size

AIM element: shape\_dimension\_representation  
Source: ISO 10303-47  
Reference Path: shape\_dimension\_representation <-  
dimensional\_characteristic\_representation.representation  
dimensional\_characteristic\_representation  
dimensional\_characteristic\_representation.dimension ->  
dimensional\_characteristic  
dimensional\_characteristic = dimensional\_size  
dimensional\_size

#2: Dimensional\_tolerance as Dimensional\_location

AIM element: shape\_dimension\_representation  
 Source: ISO 10303-47  
 Reference Path: shape\_dimension\_representation <-  
 dimensional\_characteristic\_representation.representation  
 dimensional\_characteristic\_representation  
 dimensional\_characteristic\_representation.dimension ->  
 dimensional\_characteristic  
 dimensional\_characteristic = dimensional\_location  
 dimensional\_location

### 5.1.10.10.1 dimension\_value

AIM element: measure\_representation\_item  
 Source: ISO 10303-45  
 Reference Path: shape\_dimension\_representation <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit

### 5.1.10.10.2 dimension\_description

#1 subtype is a Size\_tolerance

AIM element: (dimensional\_size.name)  
 Source: ISO 10303-47  
 ISO 10303-41  
 Reference Path: shape\_dimension\_representation <-  
 dimensional\_characteristic\_representation.representation  
 dimensional\_characteristic\_representation  
 dimensional\_characteristic\_representation.dimension ->  
 dimensional\_characteristic  
 dimensional\_characteristic = dimensional\_size  
 dimensional\_size  
 dimensional\_size.name

ISO 10303-223:2008 (E)

#2 subtype is a Location\_tolerance

AIM element: (dimensional\_size.name)  
(shape\_aspect\_relationship.description)  
Source: ISO 10303-47  
ISO 10303-41  
Reference Path: shape\_dimension\_representation <-  
dimensional\_characteristic\_representation.representation  
dimensional\_characteristic\_representation  
dimensional\_characteristic\_representation.dimension ->  
dimensional\_characteristic  
dimensional\_characteristic = dimensional\_location  
dimensional\_location<=  
shape\_aspect\_representation  
shape\_aspect\_representation.description

### 5.1.10.10.3 dimension\_note

AIM element: descriptive\_representation\_item.description  
Source: ISO 10303-43  
Reference Path: shape\_dimension\_representation<=  
shape\_representation<=  
representation  
representation.items[i] ->  
representation\_item=>  
(representation\_item.name= 'dimensional note')  
descriptive\_representation\_item  
(descriptive\_representation\_item.description= 'auxiliary')  
(descriptive\_representation\_item.description= 'theoretical')



**5.1.10.10.4 significant\_digits**

AIM element: precision\_qualifier.precision\_value  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
 Reference Path: shape\_dimension\_representation <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 {representation\_item =>  
 measure\_representation\_item}  
 representation\_item =>  
 qualified\_representation\_item  
 qualified\_representation\_item.qualifiers[1] ->  
 value\_qualifier  
 value\_qualifier = precision\_qualifier  
 precision\_qualifier  
 precision\_qualifier.precision\_value

**5.1.10.10.5 unit\_of\_measure**

AIM element: unit  
 Source: ISO 10303-41  
 Reference Path: shape\_dimension\_representation <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit  
 measure\_with\_unit.unit\_component ->  
 unit

**5.1.10.10.6 dimensional\_tolerance to Tolerance\_placement (as placement)**

AIM element: IDENTICAL MAPPING

**5.1.10.10.7 dimensional\_tolerance to tolerance\_value (as limit)**

AIM element: PATH  
 Reference Path: shape\_dimension\_representation <-  
 dimensional\_characteristic\_representation.representation  
 dimensional\_characteristic\_representation

ISO 10303-223:2008 (E)

### **5.1.10.11 Distance\_along\_curve\_tolerance**

AIM element: dimensional\_location\_with\_path  
Source: ISO 10303-47

#### **5.1.10.11.1 with\_curve\_direction**

AIM element: dimensional\_location\_with\_path.path  
Source: ISO 10303-47

#### **5.1.10.11.2 distance\_along\_curve\_tolerance\_to\_shape\_aspect (as path)**

AIM element: PATH  
Reference Path: dimensional\_location\_with\_path  
dimensional\_location\_with\_path.path ->  
shape\_aspect

### 5.1.10.12 Externally\_defined\_size\_dimension

AIM element: [externally\_defined\_dimension\_definition]  
 [dimensional\_size\_with\_path]  
 Source: ISO 10303-223  
 Reference path: [externally\_defined\_dimension\_definition<=  
 externally\_defined\_item  
 {[externally\_defined\_item.source ->  
 external\_source  
 external\_source.source\_id->  
 source\_item  
 source\_item='external dimension specification']  
 [externally\_defined\_item.item\_id ->  
 source\_item  
 source\_item='external dimension']}]  
 [dimensional\_size=>  
 dimensional\_size\_with\_path]

#2: if the externally\_defined\_size\_dimension is specified and the optional path is not specified

AIM element: [externally\_defined\_dimension\_definition]  
 [dimensional\_size]  
 Source: ISO 10303-223  
 Reference path: [externally\_defined\_dimension\_definition<=  
 externally\_defined\_item  
 {[externally\_defined\_item.source ->  
 external\_source  
 external\_source.source\_id->  
 source\_item  
 source\_item='external dimension specification']  
 [externally\_defined\_item.item\_id ->  
 source\_item  
 source\_item='external dimension']}]  
 [dimensional\_size]

#### 5.1.10.12.1 tolerance\_class

AIM element: dimensional\_size.name  
 Source: ISO 10303-47

### **5.1.10.12.2 Externally\_defined\_size\_dimension to document\_assignment (as tolerance\_definition)**

AIM element: PATH  
Reference path: externally\_defined\_dimension\_definition<-  
applied.document\_reference.items[i]  
applied.document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document  
document.description='externally size dimension specification'

### **5.1.10.12.3 externally\_defined\_size\_dimension to shape\_aspect (as path)**

#1: if the externally\_defined\_size\_dimension is specified and the optional path is specified

AIM element: PATH  
Reference path: dimensional\_size\_with\_path  
dimensional\_size\_with\_path.path ->  
shape\_aspect

#2: if the externally\_defined\_size\_dimension is specified and the optional path is not specified

AIM element: dimensional\_size  
Source: ISO 10303-47

### **5.1.10.13 Geometric\_tolerance**

AIM element: geometric\_tolerance  
Source: ISO 10303-47  
ISO 10303-519

#### **5.1.10.13.1 geometric\_tolerance\_value**

AIM element: measure\_with\_unit.value\_component  
Source: ISO 10303-41  
Reference Path: geometric\_tolerance  
geometric\_tolerance.magnitude ->  
measure\_with\_unit  
measure\_with\_unit.value\_component

**5.1.10.13.2 significant\_digits**

AIM element: precision\_qualifier.precision\_value  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
 Reference Path: geometric\_tolerance  
                   geometric\_tolerance.magnitude ->  
                   measure\_with\_unit <-  
                   measure\_qualification.qualifiers[1] ->  
                   value\_qualifier  
                   value\_qualifier = precision\_qualifier  
                   precision\_qualifier  
                   precision\_qualifier.precision\_value

**5.1.10.13.3 unit\_of\_measure**

AIM element: unit  
 Source: ISO 10303-41  
 Reference Path: geometric\_tolerance  
                   geometric\_tolerance.magnitude ->  
                   measure\_with\_unit  
                   measure\_with\_unit.unit\_component ->  
                   unit

**5.1.10.13.4 geometric\_tolerance\_to\_material\_condition\_modifier (as modifier\_control)**

AIM element: PATH  
 Reference Path: geometric\_tolerance =>  
                   modified\_geometric\_tolerance  
                   modified\_geometric\_tolerance.modifier ->  
                   limit\_condition

**5.1.10.13.5 geometric\_tolerance\_to\_shape\_aspect (as applied\_shape)**

AIM element: PATH  
 Reference Path: geometric\_tolerance  
                   geometric\_tolerance.toleranced\_shape\_aspect ->  
                   shape\_aspect

### 5.1.10.13.6 **geometric\_tolerance to tolerance\_zone (as zone\_definition)**

AIM element: PATH  
Reference Path: geometric\_tolerance <-  
tolerance\_zone.defining\_tolerance[i]  
tolerance\_zone

### 5.1.10.14 **Geometric\_tolerance\_precedence\_relationship**

AIM element: geometric\_tolerance\_relationship  
Source: ISO 10303-47  
Reference Path: {geometric\_tolerance\_relationship  
geometric\_tolerance\_relationship.name = 'precedence'}

#### 5.1.10.14.1 **geometric\_tolerance\_precedence\_relationship to geometric\_tolerance(as base\_shape\_tolerance)**

AIM element: PATH  
Reference Path: geometric\_tolerance\_precedence\_relationship <=  
shape\_aspect\_relationship  
shape\_aspect\_relationship.related\_shape\_aspect ->  
shape\_aspect <-  
geometric\_tolerance.toleranced\_shape\_aspect  
geometric\_tolerance

#### 5.1.10.14.2 **geometric\_tolerance\_precedence\_relationship to geometric\_tolerance (as pattern\_shape\_tolerance)**

AIM element: PATH  
Reference Path: geometric\_tolerance\_precedence\_relationship <=  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relate\_shape\_aspect ->  
shape\_aspect <-  
geometric\_tolerance.toleranced\_shape\_aspect  
geometric\_tolerance

### 5.1.10.15 Height\_dimension

#1: if the heighth\_dimension is specified and the optional path is specified

AIM element: dimensional\_size\_with\_path  
 Source: ISO 10303-47  
 Reference Path: dimensional\_size\_with\_path<=  
 dimensional\_size  
 dimensional\_size.name='height'

#2: if the heighth\_dimension is specified and the optional path is not specified

AIM element: dimensional\_size  
 Source: ISO 10303-47  
 Reference Path: dimensional\_size  
 dimensional\_size.name='height'

#### 5.1.10.15.1 Height\_dimension to shape\_aspect (as path)

#1: if the heighth\_dimension is specified and the optional path is specified

AIM element: PATH  
 Reference Path: dimensional\_size\_with\_path  
 dimensional\_size\_with\_path.path ->  
 shape\_aspect

#2: if the heighth\_dimension is specified and the optional path is not specified

AIM element: dimensional\_size  
 Source: ISO 10303-47

### 5.1.10.16 Length\_dimension

#1: if the length\_dimension is specified and the optional path is specified

AIM element: dimensional\_size\_with\_path  
Source: ISO 10303-47  
Reference Path: dimensional\_size\_with\_path<=  
dimensional\_size  
dimensional\_size.name='length'

#2: if the length\_dimension is specified and the optional path is not specified

AIM element: dimensional\_size  
Source: ISO 10303-47  
Reference Path: dimensional\_size  
dimensional\_size.name='length'

#### 5.1.10.16.1 Length\_dimension to shape\_aspect (as path)

#1: if the length\_dimension is specified and the optional path is specified

AIM element: PATH  
Reference Path: dimensional\_size\_with\_path  
dimensional\_size\_with\_path.path ->  
shape\_aspect

#2: if the length\_dimension is specified and the optional path is not specified

AIM element: dimensional\_size  
Source: ISO 10303-47

### 5.1.10.17 Limits\_and\_fits

#1: if the limits\_and\_fits is selected for limits\_and\_fits

AIM element: limits\_and\_fits  
Source: ISO 10303-47

#2: if the limits\_and\_fits is selected for numeric\_parameter

AIM element: (qualified\_representation\_item)  
Source: ISO 10303-45



### 5.1.10.17.1 deviation

#1: if the `limits_and_fits` is selected for `limits_and_fits`

AIM element: `limits_and_fits.form_variance`  
Source: ISO 10303-47

#2: if the `limits_and_fits` is selected for `numeric_parameter`

AIM element: `(qualitative_uncertainty.uncertainty_value)`  
Source: ISO 10303-45  
dependent\_instantiable\_uncertainty\_qualifier – (See 5.2.4.12)  
Reference Path: `{qualified_representation_item <=`  
`representation_item}`  
`qualified_representation_item`  
`qualified_representation_item.qualifiers[i] ->`  
`value_qualifier`  
`value_qualifier = uncertainty_qualifier`  
`uncertainty_qualifier.measure_name='form variance'`  
`uncertainty_qualifier =>`  
`qualitative_uncertainty`  
`qualitative_uncertainty.uncertainty_value`

### 5.1.10.17.2 fitting\_type

#1: if the `limits_and_fits` is selected for `limits_and_fits`

AIM element: `limits_and_fits.zone_variance`  
Source: ISO 10303-47

#2: if the `limits_and_fits` is selected for `numeric_parameter`

AIM element: `(qualitative_uncertainty.uncertainty_value)`  
Source: ISO 10303-45  
dependent\_instantiable\_uncertainty\_qualifier – (See 5.2.4.12)  
Reference Path: `{qualified_representation_item <=`  
`representation_item}`  
`qualified_representation_item`  
`qualified_representation_item.qualifiers[i] ->`  
`value_qualifier`  
`value_qualifier = uncertainty_qualifier`  
`uncertainty_qualifier.measure_name='zone variance'`  
`uncertainty_qualifier =>`  
`qualitative_uncertainty`  
`qualitative_uncertainty.uncertainty_value`

ISO 10303-223:2008 (E)

### 5.1.10.17.3 grade

#1: if the limits\_and\_fits is selected for limits\_and\_fits

AIM element: limits\_and\_fits.grade  
Source: ISO 10303-47

#2: if the limits\_and\_fits is selected for numeric\_parameter

AIM element: (qualitative\_uncertainty.uncertainty\_value)  
Source: ISO 10303-45  
dependent\_instantiable\_uncertainty\_qualifier – (See 5.2.4.12)  
Reference Path: {qualified\_representation\_item <=  
representation\_item}  
qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->  
value\_qualifier  
value\_qualifier = uncertainty\_qualifier  
uncertainty\_qualifier.measure\_name='grade'  
uncertainty\_qualifier =>  
qualitative\_uncertainty  
qualitative\_uncertainty.uncertainty\_value

### 5.1.10.18 Location\_dimension\_tolerance

#1: if directed = true

AIM element: (directed\_dimensional\_location)  
Source: ISO 10303-223  
Rules:  
Reference Path: #1: directed\_dimensional\_location <=  
dimensional\_location

#2: if directed = false

AIM element: (dimensional\_location)  
Source: ISO 10303-47

### 5.1.10.18.1 location\_dimension\_tolerance to orientation (as plane\_and\_direction)

#1: if directed = true

AIM element: PATH  
 Reference Path: (directed\_dimensional\_location <=  
 dimensional\_location <=)

#2: if directed = false

AIM element: PATH  
 Reference Path: (dimensional\_location <=  
 shape\_aspect\_relationship  
 shape\_definition = shape\_aspect\_relationship  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'orientation'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.10.19 Location\_tolerance

AIM element: dimensional\_location  
 Source: ISO 10303-47

#### 5.1.10.19.1 directed

AIM element: IDENTICAL MAPPING

### 5.1.10.19.2 location\_tolerance to shape\_element (as origin\_shape)

AIM element: PATH  
Reference Path: dimensional\_location <=  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect

### 5.1.10.19.3 location\_tolerance to shape\_element (as termination\_shape)

AIM element: PATH  
Reference Path: dimensional\_location <=  
shape\_aspect\_relationship  
shape\_aspect\_relationship.related\_shape\_aspect ->  
shape\_aspect

### 5.1.10.20 Material\_condition\_modifier

AIM element: limit\_condition  
Source: ISO 10303-47

#### 5.1.10.20.1 material\_type

AIM element: IDENTICAL\_MAPPING

#### 5.1.10.21 Placed\_target

AIM element: placed\_datum\_target\_feature  
Source: ISO 10303-223  
Reference Path: placed\_datum\_target\_feature <=  
datum\_target  
{datum\_target <=  
shape\_aspect  
(shape\_aspect.description = 'point')  
(shape\_aspect.description = 'line')  
(shape\_aspect.description = 'rectangle')  
(shape\_aspect.description = 'circle')}

### 5.1.10.21.1 placed\_target to orientation (as placement)

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: placed\_datum\_target\_feature <=  
 datum\_target <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'orientation'}  
 representation\_item =>  
 geometric\_representation\_item =>  
 placement

### 5.1.10.22 Plus\_minus\_value

#1: if the plus\_minus\_value is selected for tolerance\_value

AIM element: #1: (tolerance\_value)  
 Source: ISO 10303-47

#2: if the plus\_minus\_value is selected for numeric\_parameter

AIM element: (qualified\_representation\_item)  
 Source: ISO 10303-45

### 5.1.10.22.1 lower\_limit

#1: if the plus\_minus\_value is selected for tolerance\_value

AIM element: #1: (tolerance\_value.lower\_bound)  
Source: ISO 10303-47

#2: if the plus\_minus\_value is selected for numeric\_parameter

AIM element: (standard\_uncertainty.uncertainty\_value)  
Source: ISO 10303-47  
dependent\_instantiable\_uncertainty\_qualifier – (See 5.2.4.12)  
Reference Path: {qualified\_representation\_item <=  
representation\_item}  
qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->  
value\_qualifier  
value\_qualifier = uncertainty\_qualifier  
uncertainty\_qualifier.measure\_name='lower limit'  
uncertainty\_qualifier =>  
standard\_uncertainty  
standard\_uncertainty.uncertainty\_value

### 5.1.10.22.2 significant\_digits

#1: if the plus\_minus\_value is selected for tolerance\_value

AIM element: precision\_qualifier.precision\_value  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
 Reference Path: #1: (tolerance\_value  
 [tolerance\_value.upper\_bound ->]  
 [tolerance\_value.lower\_bound ->]  
 measure\_with\_unit <-  
 measure\_qualification.qualifiers[1] ->)  
 value\_qualifier  
 value\_qualifier = precision\_qualifier  
 precision\_qualifier  
 precision\_qualifier.precision\_value

#2: if the plus\_minus\_value is selected for numeric\_parameter

AIM element: precision\_qualifier.precision\_value  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
 Reference Path: (qualified\_representation\_item  
 qualified\_representation\_item.qualifiers[i] ->)  
 value\_qualifier  
 value\_qualifier = precision\_qualifier  
 precision\_qualifier  
 precision\_qualifier.precision\_value

ISO 10303-223:2008 (E)

### 5.1.10.22.3 upper\_limit

#1: if the plus\_minus\_value is selected for tolerance\_value

AIM element: (tolerance\_value.upper\_bound)  
Source: ISO 10303-47

#2: if the plus\_minus\_value is selected for numeric\_parameter

AIM element: (standard\_uncertainty.uncertainty\_value)  
Source: ISO 10303-47  
dependent\_instantiable\_uncertainty\_qualifier – (See 5.2.4.12)  
Reference Path: {qualified\_representation\_item <=  
representation\_item}  
qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->  
value\_qualifier  
value\_qualifier = uncertainty\_qualifier  
uncertainty\_qualifier.measure\_name='upper limit'  
uncertainty\_qualifier =>  
standard\_uncertainty  
standard\_uncertainty.uncertainty\_value

### 5.1.10.23 Projection

AIM element: projected\_zone\_definition  
Source: ISO 10303-47

#### 5.1.10.23.1 projection\_length

AIM element: [measure\_with\_unit.value\_component]  
[measure\_with\_unit.unit\_component]  
Source: ISO 10303-41  
Reference Path: projected\_zone\_definition  
projected\_zone\_definition.projected\_length ->  
measure\_with\_unit  
[measure\_with\_unit.value\_component]  
[measure\_with\_unit.unit\_component]

#### 5.1.10.23.2 projection to shape\_element (as projection\_end)

AIM element: PATH  
Reference Path: projected\_zone\_definition  
projected\_zone\_definition.projection\_end ->  
shape\_aspect



**5.1.10.24 Radial\_dimension\_tolerance**

AIM element: dimensional\_size  
 Source: ISO 10303-47  
 Reference Path: {dimensional\_size  
 dimensional\_size.name = 'radius'}

**5.1.10.25 Size\_tolerance**

AIM element: dimensional\_size  
 Source: ISO 10303-47

**5.1.10.25.1 envelope**

AIM element: representation.name  
 Source: ISO 10303-43  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: dimensional\_characteristic\_representation  
 dimensional\_characteristic\_representation.representation ->  
 shape\_dimension\_representation <=  
 shape\_representation <=  
 representation  
 representation.name  
 {representation.name = 'envelope tolerance'}

**5.1.10.25.2 size\_tolerance to shape\_element (as applied\_shape)**

AIM element: PATH  
 Reference Path: dimensional\_size  
 dimensional\_size.applies\_to ->  
 shape\_aspect

**5.1.10.26 Target\_area**

AIM element: datum\_target  
 Source: ISO 10303-47

**5.1.10.26.1 target\_area to shape\_element (as area\_shape)**

AIM element: IDENTICAL MAPPING

### 5.1.10.27 Target\_circle

AIM element: placed\_datum\_target\_feature  
Source: ISO 10303-223  
Reference Path: placed\_datum\_target\_feature <=  
datum\_target  
{datum\_target <=  
shape\_aspect  
shape\_aspect.description = 'circle'}

#### 5.1.10.27.1 target\_diameter

AIM element: measure\_representation\_item  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path: placed\_datum\_target\_feature <=  
datum\_target <=  
shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
property\_definition\_representation.definition  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
{representation =>  
shape\_representation =>  
shape\_representation\_with\_parameters}  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'target diameter'}  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.10.28 Target\_line

AIM element: placed\_datum\_target\_feature  
 Source: ISO 10303-223  
 Reference Path: placed\_datum\_target\_feature <=  
 datum\_target  
 {datum\_target <=  
 shape\_aspect  
 shape\_aspect.description = 'line'}

#### 5.1.10.28.1 target\_length

AIM element: measure\_representation\_item  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: placed\_datum\_target\_feature <=  
 datum\_target <=  
 shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <=  
 property\_definition.definition  
 property\_definition <=  
 property\_definition\_representation.definition  
 {property\_definition\_representation =>  
 shape\_definition\_representation}  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 {representation =>  
 shape\_representation =>  
 shape\_representation\_with\_parameters}  
 representation  
 representation.items[i] ->  
 {representation\_item  
 representation\_item.name = 'target length'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.10.29 Target\_point

AIM element: placed\_datum\_target\_feature  
Source: ISO 10303-223  
Reference Path: placed\_datum\_target\_feature <=  
datum\_target  
{datum\_target <=  
shape\_aspect  
shape\_aspect.description = 'point'}

### 5.1.10.30 Target\_rectangle

AIM element: placed\_datum\_target\_feature  
Source: ISO 10303-223  
Reference Path: placed\_datum\_target\_feature <=  
datum\_target  
{datum\_target <=  
shape\_aspect  
shape\_aspect.description = 'rectangle'}

**5.1.10.31 target\_length**

AIM element:      measure\_representation\_item  
 Source:            ISO 10303-45  
 Rules:             dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path:   placed\_datum\_target\_feature <=  
                     datum\_target <=  
                     shape\_aspect  
                     shape\_definition = shape\_aspect  
                     shape\_definition  
                     characterized\_definition = shape\_definition  
                     characterized\_definition <-  
                     property\_definition.definition  
                     property\_definition <-  
                     property\_definition\_representation.definition  
                     {property\_definition\_representation =>  
                     shape\_definition\_representation}  
                     property\_definition\_representation  
                     property\_definition\_representation.used\_representation ->  
                     {representation =>  
                     shape\_representation =>  
                     shape\_representation\_with\_parameters}  
                     representation  
                     representation.items[i] ->  
                     {representation\_item  
                     representation\_item.name = 'target length'}  
                     representation\_item =>  
                     measure\_representation\_item  
                     {measure\_representation\_item <=  
                     measure\_with\_unit =>  
                     length\_measure\_with\_unit}

### 5.1.10.31.1 target\_width

AIM element:       measure\_representation\_item  
Source:            ISO 10303-45  
Rules:             dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference Path:    placed\_datum\_target\_feature <=  
                    datum\_target <=  
                    shape\_aspect  
                    shape\_definition = shape\_aspect  
                    shape\_definition  
                    characterized\_definition = shape\_definition  
                    characterized\_definition <=  
                    property\_definition.definition  
                    property\_definition <=  
                    property\_definition\_representation.definition  
                    {property\_definition\_representation =>  
                    shape\_definition\_representation}  
                    property\_definition\_representation  
                    property\_definition\_representation.used\_representation ->  
                    {representation =>  
                    shape\_representation =>  
                    shape\_representation\_with\_parameters}  
                    representation  
                    representation.items[i] ->  
                    {representation\_item  
                    representation\_item.name = 'target width'}  
                    representation\_item =>  
                    measure\_representation\_item  
                    {measure\_representation\_item <=  
                    measure\_with\_unit =>  
                    length\_measure\_with\_unit}

### 5.1.10.32 Thickness\_tolerance

#1: if the thickness\_tolerance is specified and the optional path is specified

AIM element: dimensional\_size\_with\_path  
 Source: ISO 10303-47  
 Reference Path: dimensional\_size\_with\_path<=  
 dimensional\_size  
 dimensional\_size.name='thickness size'

#2: if the thickness\_tolerance is specified and the optional path is not specified

AIM element: dimensional\_size  
 Source: ISO 10303-47  
 Reference Path: dimensional\_size  
 dimensional\_size.name='thickness size'

#### 5.1.10.32.1 Thickness\_tolerance\_to\_shape\_aspect (as path)

#1: if the thickness\_tolerance is specified and the optional path is specified

AIM element: PATH  
 Reference Path: dimensional\_size\_with\_path  
 dimensional\_size\_with\_path.path ->  
 shape\_aspect

#2: if the thickness\_tolerance is specified and the optional path is not specified

AIM element: dimensional\_size  
 Source: ISO 10303-47

### 5.1.10.33 Tolerance\_limit

AIM element: type\_qualifier  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)

#### 5.1.10.33.1 limit\_qualifier

AIM element: type\_qualifier.name  
 Source: ISO 10303-45  
 Rules: dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
 Reference Path: (type\_qualifier.name='maximum')  
 (type\_qualifier.name='minimum')  
 (type\_qualifier.name)

### 5.1.10.34 Tolerance\_placement

AIM element: shape\_dimension\_representation  
Source: ISO 10303-457

#### 5.1.10.34.1 applied\_direction

AIM element: direction  
Source: ISO 10303-45  
Reference Path: shape\_dimension\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
representation\_item =>  
(representation\_item.name='applied direction')  
geometric\_representation\_item=>  
direction

#### 5.1.10.34.2 origin\_point

AIM element: cartesian\_point  
Source: ISO 10303-45  
Reference Path: shape\_dimension\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
representation\_item =>  
(representation\_item.name='origin point')  
geometric\_representation\_item=>  
point=>  
cartesian\_point

#### 5.1.10.34.3 termination\_point

AIM element: cartesian\_point  
Source: ISO 10303-45  
Reference Path: shape\_dimension\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
representation\_item =>  
(representation\_item.name='termination point')  
geometric\_representation\_item=>  
point=>  
cartesian\_point



### 5.1.10.34.4 Tolerance\_range

#1: if the tolerance\_range is selected for tolerance\_value

AIM element: (shape\_dimension\_representation)  
Source: ISO 10303-47

#2: if the tolerance\_range is selected for numeric\_parameter

AIM element: ([value\_range]  
[qualified\_representation\_item])  
Source: ISO 10303-223  
ISO 10303-45  
Reference Path: {value\_range <=  
compound\_representation\_item  
compound\_representation\_item.item\_element ->  
set\_representation\_item  
set\_representation\_item[i] ->  
representation\_item =>  
measure\_representation\_item <=)  
measure\_with\_unit  
measure\_with\_unit.unit\_component ->  
unit}

### 5.1.10.34.5 lower\_range

#1: if the tolerance\_range is selected for tolerance\_value

AIM element: measure\_representation\_item  
Source: ISO 10303-45  
Reference Path: shape\_dimension\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'lower range'}}  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.value\_component

ISO 10303-223:2008 (E)

#2: if the tolerance\_range is selected for numeric\_parameter

AIM element: measure\_with\_unit.value\_component  
Source: ISO 10303-41  
Reference Path: value\_range <=  
compound\_representation\_item  
compound\_representation\_item.item\_element ->  
set\_representation\_item  
set\_representation\_item[i] ->  
representation\_item =>  
{representation\_item.name = 'lower limit'}  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.value\_component

### 5.1.10.34.6 significant\_digits

#1: if the tolerance\_range is selected for tolerance\_value

AIM element: precision\_qualifier.precision\_value  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
Reference Path: shape\_dimension\_representation <==  
shape\_representation <=  
representation  
representation.items[i] ->  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit <-  
measure\_qualification.qualified\_measure  
measure\_qualification  
measure\_qualification.qualifiers[i] ->  
value\_qualifier = precision\_qualifier  
precision\_qualifier  
precision\_qualifier.precision\_value

#2: if the tolerance\_range is selected for numeric\_parameter

AIM element: precision\_qualifier.precision\_value  
Source: ISO 10303-45  
Rules: dependent\_instantiable\_precision\_qualifier – (See 5.2.4.10)  
Reference Path: (qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->)  
value\_qualifier  
value\_qualifier = precision\_qualifier  
precision\_qualifier  
precision\_qualifier.precision\_value

**5.1.10.34.7 upper\_range**

#1: if the tolerance\_range is selected for tolerance\_value

AIM element:      measure\_with\_unit.value\_component  
 Source:            ISO 10303-45  
 Reference Path:    shape\_dimension\_representation <=  
                     representation  
                     representation.items[i] ->  
                     {representation\_item  
                     representation\_item.name = 'upper range'}  
                     measure\_representation\_item<=  
                     measure\_with\_unit  
                     measure\_with\_unit.value\_component

#2: if the tolerance\_range is selected for numeric\_parameter

AIM element:      measure\_with\_unit.value\_component  
 Source:            ISO 10303-41  
 Reference Path:    value\_range <=  
                     compound\_representation\_item  
                     compound\_representation\_item.item\_element ->  
                     set\_representation\_item  
                     set\_representation\_item[i] ->  
                     representation\_item =>  
                     {representation\_item.name = 'upper limit'}}  
                     measure\_representation\_item<=  
                     measure\_with\_unit  
                     measure\_with\_unit.value\_component

**5.1.10.35 Tolerance\_value**

AIM element:      dimensional\_characteristic\_representation  
 Source:            ISO 10303-45

### 5.1.10.35.1 tolerance\_value to limits\_and\_fits (as defined\_value)

AIM element: PATH  
Source: ISO 10303-522  
Reference Path: dimensional\_characteristic\_representation  
dimensional\_characteristic\_representation.dimension ->  
dimensional\_characteristic <-  
plus\_minus\_tolerance.toleranced\_dimension  
plus\_minus\_tolerance  
plus\_minus\_tolerance.range ->  
tolerance\_method\_definition  
tolerance\_method\_definition = limits\_and\_fits  
limits\_and\_fits

### 5.1.10.35.2 tolerance\_value to plus\_minus\_value (as defined\_value)

AIM element: PATH  
Reference Path: dimensional\_characteristic\_representation  
dimensional\_characteristic\_representation.dimension ->  
dimensional\_characteristic <-  
plus\_minus\_tolerance.toleranced\_dimension  
plus\_minus\_tolerance  
plus\_minus\_tolerance.range ->  
tolerance\_method\_definition  
tolerance\_method\_definition = tolerance\_value  
tolerance\_value

### 5.1.10.35.3 tolerance\_value to tolerance\_limit (as defined\_value)

AIM element: PATH  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
dependent\_instantiable\_type\_qualifier – (See 5.2.4.11)  
Reference Path: dimensional\_characteristic\_representation  
dimensional\_characteristic\_representation.representation ->  
shape\_dimension\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
{representation\_item =>  
measure\_representation\_item}  
representation\_item =>  
qualified\_representation\_item  
qualified\_representation\_item.qualifiers[i] ->  
value\_qualifier  
value\_qualifier = type\_qualifier  
type\_qualifier

**5.1.10.35.4 tolerance\_value to tolerance\_range (as defined\_value)**

AIM element: PATH  
 Reference Path: dimensional\_characteristic\_representation  
 dimensional\_characteristic\_representation.representation ->  
 shape\_dimension\_representation

**5.1.10.36 Tolerance\_zone**

AIM element: tolerance\_zone  
 Source: ISO 10303-47

**5.1.10.36.1 common\_zone**

AIM element: tolerance\_zone.defining\_tolerance  
 Source: ISO 10303-47

**5.1.10.36.2 form\_type**

AIM element: tolerance\_zone\_form.name  
 Source: ISO 10303-47  
 Reference Path: tolerance\_zone  
 tolerance\_zone.form ->  
 tolerance\_zone\_form  
 tolerance\_zone\_form.name

**5.1.10.36.3 tolerance\_zone to projection (as extended\_shape)**

AIM element: PATH  
 Reference Path: tolerance\_zone <-  
 tolerance\_zone\_definition.zone  
 tolerance\_zone\_definition =>  
 projected\_zone\_definition

**5.1.10.36.4 tolerance\_zone to tolerance\_zone\_definition (as zone\_definition)**

AIM element: PATH  
 Reference Path: tolerance\_zone <-  
 tolerance\_zone\_definition.zone  
 tolerance\_zone\_definition

**5.1.10.37 Tolerance\_zone\_definition**

AIM element: tolerance\_zone\_definition  
 Source: ISO 10303-47

### 5.1.10.37.1 tolerance\_zone\_definition to shape\_element (as first\_element)

AIM element: PATH  
Reference Path: tolerance\_zone\_definition  
tolerance\_zone\_definition.boundaries[i] ->  
shape\_aspect

### 5.1.10.37.2 tolerance\_zone\_definition to shape\_element (as second\_element)

AIM element: PATH  
Reference Path: tolerance\_zone\_definition  
tolerance\_zone\_definition.boundaries[i] ->  
shape\_aspect

### 5.1.10.38 Width\_dimension

#1: if the length\_dimension is specified and the optional path is specified

AIM element: dimensional\_size\_with\_path  
Source: ISO 10303-47  
Reference Path: dimensional\_size\_with\_path<=  
dimensional\_size  
dimensional\_size.name='width'

## 5.1.11 Manufacturing\_feature UoF

### 5.1.11.1 Compound\_feature

AIM element: compound\_feature  
Source: ISO 10303-522  
Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference path: compound\_feature <=  
{feature\_definition=>  
instanced\_feature}  
feature\_definition<=  
characterized\_object  
characterized\_definition=characterized\_object<-  
property\_definition.definition  
property\_definition  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect  
shape\_aspect.name = 'compound feature in solid'

### 5.1.11.1.1 feature\_description

AIM element: characterized\_object.description  
Source: ISO 10303-41  
Reference path: compound\_feature <=  
feature\_definition <=  
characterized\_object  
characterized\_object.description

### 5.1.11.1.2 feature\_name

AIM element: characterized\_object.name  
Source: ISO 10303-41  
Reference path: compound\_feature <=  
feature\_definition <=  
characterized\_object  
characterized\_object.name

### 5.1.11.1.3 compound\_feature to compound\_feature\_element (as element)

#1: as element except for thread

AIM element: PATH  
Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
transition\_feature\_life\_cycle – (See 5.2.4.31)  
transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
Reference path: compound\_feature <=  
feature\_definition <=  
characterized\_object  
characterized\_definition=characterized\_object<-  
property\_definition.definition  
property\_definition  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect  
{shape\_aspect=>  
composite\_shape\_aspect}  
shape\_aspect<-  
shape\_aspect\_relationship.relating\_shape\_aspect  
shape\_aspect\_relationship  
{shape\_aspect\_relationship=>  
feature\_component\_relationship}  
shape\_aspect\_relationship.related\_shape\_aspect->  
shape\_aspect=>  
(( instanced\_feature)  
( transition\_feature))



#2: as element for features using applied area only

AIM element: PATH

Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
 transition\_feature\_life\_cycle – (See 5.2.4.31)  
 transition\_feature\_on\_part\_boundary – (See 5.2.4.32)

Reference path: compound\_feature <=  
 feature\_definition <=  
 characterized\_object  
 characterized\_definition=characterized\_object<-  
 property\_definition.definition  
 property\_definition  
 property\_definition=>  
 product\_definition\_shape<-  
 shape\_aspect.of\_shape  
 shape\_aspect  
 {shape\_aspect=>  
 composite\_shape\_aspect}  
 shape\_aspect<-  
 shape\_aspect\_relationship.relating\_shape\_aspect  
 shape\_aspect\_relationship  
 {shape\_aspect\_relationship=>  
 feature\_component\_relationship}  
 shape\_aspect\_relationship.related\_shape\_aspect->  
 shape\_aspect=>  
 (applied\_area)

### 5.1.11.2 Compound\_feature\_element

#1: as element except for thread

AIM element: (instanced\_feature)  
(transition\_feature)  
#2(applied\_area)  
Source: ISO 10303-522  
Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
transition\_feature\_life\_cycle – (See 5.2.4.31)  
transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
Reference path: (instanced\_feature <=  
[feature\_definition <=  
characterized\_object]  
[shape\_aspect])  
(transition\_feature <=  
shape\_aspect)

#2: as element for thread only

AIM element: (applied\_area)  
Source: ISO 10303-522  
Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
transition\_feature\_life\_cycle – (See 5.2.4.31)  
transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
Reference path: (applied\_area <=  
shape\_aspect)

#### 5.1.11.2.1 compound\_feature\_element to machining\_feature (as element)

AIM element: IDENTICAL MAPPING

#### 5.1.11.2.2 compound\_feature\_element to transition\_feature (as element)

AIM element: IDENTICAL MAPPING

### 5.1.11.3 Compound\_feature\_relationship

AIM element: shape\_aspect\_relationship  
Source: ISO 10303-41  
Reference path: {shape\_aspect\_relationship  
shape\_aspect\_relationship.name = 'compound feature ordering'}

### 5.1.11.3.1 compound\_feature\_relationship to compound\_feature\_element (as predecessor)

AIM element: PATH  
 Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
 transition\_feature\_life\_cycle – (See 5.2.4.31)  
 transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
 Reference path: shape\_aspect\_relationship  
 shape\_aspect\_relationship.related\_shape\_aspect ->  
 shape\_aspect =>  
 ( instanced\_feature)  
 ( transition\_feature)

### 5.1.11.3.2 compound\_feature\_relationship to compound\_feature\_element (as successor)

AIM element: PATH  
 Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
 transition\_feature\_life\_cycle – (See 5.2.4.31)  
 transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
 Reference path: shape\_aspect\_relationship  
 shape\_aspect\_relationship.relying\_shape\_aspect ->  
 shape\_aspect =>  
 ( instanced\_feature)  
 ( transition\_feature)

### 5.1.11.4 Feature

AIM element: characterized\_object  
 Source: ISO 10303-41

### 5.1.11.5 Manufacturing\_feature

AIM element: characterized\_object  
 Source: ISO 10303-41

### 5.1.11.5.1 required\_geometry

AIM element: advanced\_brep\_shape\_representation  
Source: ISO 10303-514  
Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
Reference path: shape\_aspect  
shape\_definition = shape\_aspect  
shape\_definition  
characterized\_definition = shape\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition  
represented\_definition = property\_definition  
represented\_definition <-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation =>  
(shape\_representation)  
{representation.items[i] ->  
representation\_item =>  
(geometric\_representation\_item)  
(topological\_representation\_item)}  
representation =>  
shape\_representation=>  
advanced\_brep\_shape\_representation

### 5.1.11.6 Manufacturing\_design\_feature

AIM element: characterized\_object  
Source: ISO 10303-41

### 5.1.11.7 Machining\_feature

AIM element: instanced\_feature  
Source: ISO 10303-522  
Rules: machining\_feature\_life\_cycle – (See 5.2.4.18)  
subtype\_mandatory\_characterized\_object – (See 5.2.4.30)  
Reference path: instanced\_feature <=  
[shape\_aspect]  
[feature\_definition <=  
characterized\_object]

**5.1.11.7.1 usage\_name**

AIM element: shape\_aspect.description  
 Source: ISO 10303-41  
 Reference Path: instanced\_feature <=  
                   shape\_aspect  
                   shape\_aspect.description

**5.1.11.7.2 machining\_feature to orientation (as placement)**

AIM element: PATH  
 Rules: dependent\_instantiable\_shape\_representation – (See 5.2.4.14)  
 Reference Path: instanced\_feature <=  
                   shape\_aspect  
                   shape\_definition = shape\_aspect  
                   shape\_definition  
                   characterized\_definition = shape\_definition  
                   characterized\_definition <=  
                   property\_definition.definition  
                   {property\_definition=>  
                   product\_definition\_shape}  
                   property\_definition <=  
                   property\_definition\_representation.definition  
                   {property\_definition\_representation =>  
                   shape\_definition\_representation}  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   {representation =>  
                   shape\_representation =>  
                   shape\_representation\_with\_parameters}  
                   representation  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'orientation'}  
                   representation\_item =>  
                   geometric\_representation\_item =>  
                   placement

**5.1.11.8 Manufacturing\_feature\_group**

AIM element: group  
 Source: ISO 10303-41

ISO 10303-223:2008 (E)

### 5.1.11.8.1 group\_description

AIM element: group.description  
Source: ISO 10303-41

### 5.1.11.8.2 group\_name

AIM element: group.name  
Source: ISO 10303-41

### 5.1.11.8.3 manufacturing\_feature\_group to manufacturing\_feature (as feature\_groups)

AIM element: PATH  
Reference path: group<-  
group\_assignment.assigned\_group  
group\_assignment=>  
applied\_group\_assignment  
applied\_group\_assignment.items->  
group\_item  
(group\_item=instanced\_feature)  
(group\_item=replicate\_feature)  
(group\_item=transition\_feature)

### 5.1.11.8.4 manufacturing\_feature\_group to manufacturing\_feature\_group (as feature\_groups)

AIM element: PATH  
Reference path: group<-  
group\_relationship.related\_group  
group\_relationship  
group\_relationship.relatng\_group->  
group

### 5.1.11.9 Transition\_feature

AIM element: transition\_feature  
Source: ISO 10303-522  
Rules: transition\_feature\_life\_cycle – (See 5.2.4.31)  
transition\_feature\_on\_part\_boundary – (See 5.2.4.32)  
Reference path: transition\_feature <=  
shape\_aspect

## 5.1.12 Manufacturing\_process\_control\_documentation UoF

### 5.1.12.1 Customer\_order

AIM element: directed\_action  
 Source: ISO 10303-41  
 Rules: customer\_order\_rules – (See 5.2.4.6)  
 Reference Path: {directed\_action <=  
 executed\_action <=  
 action  
 action.name = 'customer order'}

#### 5.1.12.1.1 customer\_order to date (as delivery\_date)

AIM element: date  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: directed\_action  
 date\_item = directed\_action  
 date\_item <-  
 applied\_date\_assignment.items[i]  
 applied\_date\_assignment <=  
 {date\_assignment  
 date\_assignment.role ->  
 date\_role  
 date\_role.name = 'delivery date'}  
 date\_assignment  
 date\_assignment.assigned\_date ->  
 date

#### 5.1.12.1.2 material\_disposition

AIM element: action.description  
 Source: ISO 10303-41  
 Reference Path: directed\_action <=  
 executed\_action <=  
 action  
 action.description

### 5.1.12.1.3 order\_number

AIM element: action\_directive.name  
Source: ISO 10303-41  
Reference Path: directed\_action  
directed\_action.directive ->  
action\_directive  
action\_directive.name

### 5.1.12.1.4 order\_status

AIM element: action\_status  
Source: ISO 10303-41  
Reference Path: directed\_action <=  
executed\_action <=  
action <-  
action\_status.assigned\_action  
action\_status

### 5.1.12.1.5 production\_status

AIM element: PATH  
Reference Path: directed\_action<=  
executed\_action<=  
action  
characterized\_action\_definition=action  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='production status'  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'current')  
(descriptive\_representation\_item.description = 'experimental')  
(descriptive\_representation\_item.description = 'prototype')  
(descriptive\_representation\_item.description = 'new')}



**5.1.12.1.6 special\_instructions**

AIM element: action\_directive.comment  
 Source: ISO 10303-41  
 Reference Path: directed\_action  
                   directed\_action.directive ->  
                   action\_directive  
                   action\_directive.comment

**5.1.12.1.7 customer\_order to first\_article (as first\_article\_requirements)**

AIM element: PATH  
 Reference Path: directed\_action  
                   directed\_action.directive->  
                   action\_directive  
                   action\_directive.requests[i]->  
                   versioned\_action\_request=>  
                   casting\_verification=>  
                   first\_article

**5.1.12.1.8 customer\_order to inspection\_plan (as inspection\_procedure)**

AIM element: PATH  
 Reference Path: directed\_action<=  
                   executed\_action<=  
                   action<-  
                   action\_relationship.related\_action  
                   action\_relationship  
                   action\_relationship.relating\_action  
                   action=>  
                   product\_definition\_process=>  
                   inspection\_plan

**5.1.12.1.9 customer\_order to request\_for\_quotation (as RFQ)**

AIM element: PATH  
 Reference Path: directed\_action  
                   directed\_action.directive->  
                   action\_directive  
                   action\_directive.requests[i]->  
                   versioned\_action\_request=>  
                   request\_for\_quotation

### 5.1.12.1.10 customer\_order to ordered\_part (as quantity\_ordered)

AIM element: PATH  
Reference Path: directed\_action <=  
executed\_action <=  
action <=  
action\_assignment.assigned\_action  
action\_assignment =>  
ordered\_part

### 5.1.12.1.11 customer\_order to casting\_process (as casting\_method)

AIM element: PATH  
 Reference Path: directed\_action<=  
 executed\_action<=  
 action  
 characterized\_action\_definition=action  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='casting method'  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.name='core presentation')  
 (descriptive\_representation\_item.name='core setting')  
 (descriptive\_representation\_item.name='die assembly')  
 (descriptive\_representation\_item.name='die handling')  
 (descriptive\_representation\_item.name='die temperature control')  
 (descriptive\_representation\_item.name='furnace loading')  
 (descriptive\_representation\_item.name='gate system removal')  
 (descriptive\_representation\_item.name='grinding')  
 (descriptive\_representation\_item.name='inspection')  
 (descriptive\_representation\_item.name='ladle loading')  
 (descriptive\_representation\_item.name='loading injection system')  
 (descriptive\_representation\_item.name='melt raw material')  
 (descriptive\_representation\_item.name='mould closing')  
 (descriptive\_representation\_item.name='mould making process')  
 (descriptive\_representation\_item.name='package and shipping')  
 (descriptive\_representation\_item.name='part routing')  
 (descriptive\_representation\_item.name='tool handling')  
 (descriptive\_representation\_item.name='pattern handling')  
 (descriptive\_representation\_item.name='pouring process')  
 (descriptive\_representation\_item.name='refractor removal')  
 (descriptive\_representation\_item.name='removal from die')  
 (descriptive\_representation\_item.name='shakeout')  
 (descriptive\_representation\_item.name='slurry process')  
 (descriptive\_representation\_item.name='sand removal')  
 (descriptive\_representation\_item.name='wax removal') }

### 5.1.12.1.12 customer\_order to non\_permanent\_moulding\_process (as casting\_method)

AIM element: PATH  
Reference Path: directed\_action<=  
executed\_action<=  
action  
characterized\_action\_definition=action  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='casting method'  
descriptive\_representation\_item  
{(descriptive\_representation\_item.name.description = 'shell')  
(descriptive\_representation\_item.name.description = 'green sand')  
(descriptive\_representation\_item.name.description = 'no bake')  
(descriptive\_representation\_item.name.description = 'lost form')  
(descriptive\_representation\_item.name.description = 'vacuum')}

### 5.1.12.1.13 customer\_order to permanent\_moulding\_process (as casting\_method)

AIM element: PATH  
 Reference Path: directed\_action<=  
 executed\_action<=  
 action  
 characterized\_action\_definition=action  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='casting method'  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.name.description = 'static')  
 (descriptive\_representation\_item.name.description = 'tilt')  
 (descriptive\_representation\_item.name.description = 'low pressure')}

### 5.1.12.1.14 customer\_order to person\_in\_organization (as customer)

AIM element: PATH  
 Reference Path: directed\_action  
 directed\_action.directive ->  
 action\_directive  
 person\_and\_organization\_item = action\_directive  
 person\_and\_organization\_item<-  
 applied\_person\_and\_organization\_assignment.items[i]  
 applied\_person\_and\_organization\_assignment<=  
 person\_and\_organization\_assignment  
 person\_and\_organization\_assignment.assigned\_person\_and\_organization ->  
 person\_and\_organization

### 5.1.12.1.15 customer\_order to project\_order (as initiated\_order)

AIM element: PATH  
Source: ISO 10303-233  
Rules: project\_order\_tracking\_relationship – (See 5.2.4.25)  
Reference Path: directed\_action <=  
executed\_action <=  
action <-  
action\_relationship.related\_action  
action\_relationship  
action\_relationship.relying\_action ->  
{action  
action.name = 'project order'}  
action =>  
executed\_action =>  
directed\_action

### 5.1.12.1.16 customer\_order to simulation\_report (as simulation\_run\_result)

AIM element: PATH  
Reference Path: directed\_action <=  
executed\_action <=  
action  
document\_reference\_item=action  
document\_reference\_item <-  
applied\_document\_reference.items[i]  
applied\_document\_reference <=  
document\_reference  
document\_reference.assigned\_document ->  
document  
document.name = 'simulation report'

### 5.1.12.2 Digital\_technical\_data\_package\_work\_order

AIM element: directed\_action  
Source: ISO 10303-41  
Reference Path: {directed\_action <=  
executed\_action <=  
action  
action.name = 'digital technical data package work order'}

### 5.1.12.2.1 order\_id

AIM element: action\_directive.name  
 Source: ISO 10303-41  
 Reference Path: directed\_action  
                   directed\_action.directive ->  
                   action\_directive  
                   action\_directive.name

### 5.1.12.3 Ordered\_part

AIM element: ordered\_part  
 Source: ISO 10303-223  
 Reference Path: ordered\_part <=  
                   [action\_assignment]  
                   [characterized\_object]

#### 5.1.12.3.1 quantity\_required

AIM element: measure\_with\_unit.value\_component  
 Source: ISO 10303-41  
 Reference Path: ordered\_part <=  
                   characterized\_object  
                   characterized\_definition = characterized\_object  
                   characterized\_definition <=  
                   property\_definition.definition  
                   property\_definition <=  
                   property\_definition\_representation.definition  
                   property\_definition\_representation  
                   property\_definition\_representation.used\_representation ->  
                   representation  
                   representation.items[i] ->  
                   representation\_item =>  
                   measure\_representation\_item <=  
                   measure\_with\_unit  
                   measure\_with\_unit.value\_component  
                   {measure\_with\_unit.value\_component ->  
                   measure\_value  
                   measure\_value = count\_measure  
                   count\_measure}

### 5.1.12.3.2 quantity\_unit\_of\_measure

AIM element: measure\_with\_unit.unit\_component  
Source: ISO 10303-41  
Reference Path: ordered\_part <=  
characterized\_object  
characterized\_definition = characterized\_object  
characterized\_definition <=  
property\_definition.definition  
property\_definition <=  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.unit\_component

### 5.1.12.4 Pedigree\_creation\_order

AIM element: directed\_action  
Source: ISO 10303-41  
Reference Path: {directed\_action <=  
executed\_action <=  
action  
action.name = 'pedigree creation order'}

#### 5.1.12.4.1 order\_id

AIM element: action\_directive.name  
Source: ISO 10303-41  
Reference Path: directed\_action  
directed\_action.directive ->  
action\_directive  
action\_directive.name

### 5.1.12.5 Project\_order

AIM element: directed\_action  
Source: ISO 10303-41  
Reference Path: {directed\_action <=  
executed\_action <=  
action  
action.name = 'project order'}



### 5.1.12.5.1 project\_order\_id

AIM element: action\_directive.name  
 Source: ISO 10303-41  
 Reference Path: directed\_action  
                   directed\_action.directive ->  
                   action\_directive  
                   action\_directive.name

### 5.1.12.5.2 project\_order to approval (as release\_authorization)

AIM element: PATH  
 Rules: project\_order\_requires\_approval – (See 5.2.4.25)  
 Reference Path: directed\_action  
                   approval\_item = directed\_action  
                   approval\_item <-  
                   applied\_approval\_assignment.items[i]  
                   applied\_approval\_assignment <=  
                   approval\_assignment  
                   approval\_assignment.assigned\_approval ->  
                   approval

### 5.1.12.5.3 project\_order to digital\_technical\_data\_packagework\_order (as technical\_data\_package\_status)

AIM element: PATH  
 Rules: project\_order\_tracking\_relationship – (See 5.2.4.25)  
 Reference Path: directed\_action <=  
                   executed\_action <=  
                   action <-  
                   action\_relationship.relatng\_action  
                   action\_relationship  
                   action\_relationship.related\_action ->  
                   {action  
                   action.name = 'digital technical data package work order'}  
                   action =>  
                   executed\_action =>  
                   directed\_action

#### 5.1.12.5.4 project\_order to pedigree\_creation\_order (as pedigree\_creation\_status)

AIM element: PATH  
Rules: project\_order\_tracking\_relationship – (See 5.2.4.25)  
Reference Path: directed\_action <=  
executed\_action <=  
action <-  
action\_relationship.relating\_action  
action\_relationship  
action\_relationship.related\_action ->  
{action  
action.name = 'pedigree creation order'}  
action =>  
executed\_action =>  
directed\_action

#### 5.1.12.5.5 project\_order to part (as part\_status)

AIM element: PATH  
Rules: part\_requires\_project\_order – (See 5.2.4.25)  
Reference Path: directed\_action <=  
executed\_action <=  
action <-  
action\_assignment.assigned\_action  
action\_assignment =>  
applied\_action\_assignment  
applied\_action\_assignment.items[i] ->  
action\_item  
action\_item = product\_definition\_formation  
product\_definition\_formation

#### 5.1.12.5.6 project\_order to requisition (as ordered\_resource)

AIM element: PATH  
Reference Path: directed\_action  
document\_reference\_item = directed\_action  
document\_reference\_item <-  
applied\_document\_reference.items[i]  
applied\_document\_reference <=  
document\_reference  
document\_reference.assigned\_document ->  
document

### 5.1.12.5.7 project\_order to resource\_acquisition\_order (as resource\_acquisition\_status)

AIM element: PATH  
 Rules: project\_order\_tracking\_relationship – (See 5.2.4.25)  
 Reference Path: directed\_action <=  
 executed\_action <=  
 action <=  
 action\_relationship.relateing\_action  
 action\_relationship  
 action\_relationship.related\_action ->  
 {action  
 action.name = 'resource acquisition order'}  
 action =>  
 executed\_action =>  
 directed\_action

### 5.1.12.5.8 project\_order to shop\_work\_order (as shop\_work\_status)

AIM element: PATH  
 Rules: project\_order\_tracking\_relationship – (See 5.2.4.25)  
 Reference Path: directed\_action <=  
 executed\_action <=  
 action <=  
 action\_relationship.relateing\_action  
 action\_relationship  
 action\_relationship.related\_action ->  
 {action  
 action.name = 'shop work order'}  
 action =>  
 executed\_action =>  
 directed\_action

### 5.1.12.6 Request\_for\_quotation

AIM element: request\_for\_quotation  
 Source: ISO 10303-223  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request

### 5.1.12.6.1 RFQ\_number

AIM element: version\_action\_request.id  
Source: ISO 10303-45  
Reference Path: request\_for\_quotation =>  
version\_action\_request  
version\_action\_request.id

### 5.1.12.6.2 quantity\_breaks

AIM element: version\_action\_request.id  
Source: ISO 10303-45  
Reference Path: request\_for\_quotation =>  
version\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='quantity breaks'  
measure\_representation\_item

### 5.1.12.6.3 quantity\_per\_release

AIM element: version\_action\_request.id  
 Source: ISO 10303-45  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request<-  
 action\_request\_solution.request  
 action\_request\_solution  
 action\_request\_solution.method->  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='quantity per release'  
 measure\_representation\_item

### 5.1.12.6.4 quantity\_per\_month

AIM element: version\_action\_request.id  
 Source: ISO 10303-45  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request<-  
 action\_request\_solution.request  
 action\_request\_solution  
 action\_request\_solution.method->  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='quantity per month'  
 measure\_representation\_item

### 5.1.12.6.5 quantity\_per\_year

AIM element: version\_action\_request.id  
Source: ISO 10303-45  
Reference Path: request\_for\_quotation =>  
version\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='quantity per year'  
measure\_representation\_item

### 5.1.12.6.6 total\_quantity\_per\_order

AIM element: version\_action\_request.id  
Source: ISO 10303-45  
Reference Path: request\_for\_quotation =>  
version\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='total quantity per order'  
measure\_representation\_item

### 5.1.12.6.7 type\_of\_quotation

AIM element: version\_action\_request.id  
 Source: ISO 10303-45  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request<-  
 action\_request\_solution.request  
 action\_request\_solution  
 action\_request\_solution.method->  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='type of quotation'  
 descriptive\_representation\_item

### 5.1.12.6.8 request\_for\_quotation\_to\_date (as delivery\_date)

AIM element: PATH  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request<-  
 date\_item=version\_action\_request  
 date\_item<-  
 applied\_date\_assignment.items[i]  
 applied\_date\_assignment<=  
 date\_assignment  
 {date\_assignment.role->  
 date\_role  
 date\_role.name='delivery date'}  
 date\_assignment.assigned\_data->  
 date

### 5.1.12.6.9 request\_for\_quotation to date (as reply\_due\_date)

AIM element: PATH  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: request\_for\_quotation =>  
version\_action\_request<-  
date\_item=version\_action\_request  
date\_item<-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment<=  
date\_assignment  
{date\_assignment.role->  
date\_role  
date\_role.name='reply due date'}  
date\_assignment.assigned\_data->  
date

### 5.1.12.6.10 request\_for\_quotation to date (as request\_date)

AIM element: PATH  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: request\_for\_quotation =>  
version\_action\_request<-  
date\_item=version\_action\_request  
date\_item<-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment<=  
date\_assignment  
{date\_assignment.role->  
date\_role  
date\_role.name='request date'}  
date\_assignment.assigned\_data->  
date



### 5.1.12.6.11 request\_for\_quotation to date (as order\_completion)

AIM element: PATH  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: request\_for\_quotation =>  
 version\_action\_request<-  
 date\_item=version\_action\_request  
 date\_item<-  
 applied\_date\_assignment.items[i]  
 applied\_date\_assignment<=  
 date\_assignment  
 {date\_assignment.role->  
 date\_role  
 date\_role.name='order completion'}  
 date\_assignment.assigned\_data->  
 date

### 5.1.12.7 Resource\_acquisition\_order

AIM element: directed\_action  
 Source: ISO 10303-41  
 Reference Path: {directed\_action <=  
 executed\_action <=  
 action  
 action.name = 'resource acquisition order'}

#### 5.1.12.7.1 order\_id

AIM element: action\_directive.name  
 Source: ISO 10303-41  
 Reference Path: directed\_action  
 directed\_action.directive ->  
 action\_directive  
 action\_directive.name

### 5.1.12.8 Shop\_work\_order

AIM element: directed\_action  
 Source: ISO 10303-41  
 Reference Path: {directed\_action <=  
 executed\_action <=  
 action  
 action.name = 'shop work order'}

ISO 10303-223:2008 (E)

### **5.1.12.8.1 order\_id**

AIM element: action\_directive.name  
Source: ISO 10303-41  
Reference Path: directed\_action  
directed\_action.directive ->  
action\_directive  
action\_directive.name

## **5.1.13 Part\_administration\_data UoF**

### **5.1.13.1 Address**

AIM element: address  
Source: ISO 10303-41

#### **5.1.13.1.1 city**

AIM element: address.town  
Source: ISO 10303-41

#### **5.1.13.1.2 email**

AIM element: address.electronic\_mail\_address  
Source: ISO 10303-41

#### **5.1.13.1.3 fax**

AIM element: address.facsimile\_number  
Source: ISO 10303-41

#### **5.1.13.1.4 state**

AIM element: address.region  
Source: ISO 10303-41

#### **5.1.13.1.5 street\_address**

AIM element: address.street\_number  
Source: ISO 10303-41

**5.1.13.1.6 web\_url**

AIM element: address.url  
 Source: ISO 10303-41

**5.1.13.1.7 zip\_code**

AIM element: address.postal\_code  
 Source: ISO 10303-41

**5.1.13.2 Approval**

AIM element: approval  
 Source: ISO 10303-41

**5.1.13.2.1 status**

AIM element: approval\_status.name  
 Source: ISO 10303-41  
 Rules: restrict\_approval\_status – (See 5.2.4.26)  
 dependent\_instantiable\_approval\_status (see 5.2.4.7)  
 Reference Path: approval  
 approval.status ->  
 approval\_status  
 approval\_status.name  
 {(approval\_status.name = 'approved')  
 (approval\_status.name = 'not yet approved')  
 (approval\_status.name = 'disapproved')  
 (approval\_status.name = 'withdrawn')}

**5.1.13.2.2 approval to date (as approval\_date)**

AIM element: date  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: approval <-  
 approval\_date\_time.dated\_approval  
 approval\_date\_time  
 approval\_date\_time.date\_time ->  
 date\_time\_select  
 date\_time\_select = date  
 date

### 5.1.13.2.3 approval to person\_in\_organization (as approval\_authority)

AIM element: PATH  
Reference Path: approval <-  
approval\_person\_organization.authorized\_approval  
approval\_person\_organization  
approval\_person\_organization.person\_organization ->  
person\_organization\_select  
person\_organization\_select = person\_and\_organization  
person\_and\_organization

### 5.1.13.3 Date

AIM element: date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )

### 5.1.13.4 Organization

AIM element: organization  
Source: ISO 10303-41

#### 5.1.13.4.1 division

AIM element: organization\_address.description  
Source: ISO 10303-41  
Path: organization<-  
organizational\_address.organizations[i]  
organizational\_address  
organizational\_address.description

#### 5.1.13.4.2 organization\_id

AIM element: organization.id  
Source: ISO 10303-41

#### 5.1.13.4.3 organization\_name

AIM element: organization.name  
Source: ISO 10303-41

**5.1.13.4 organization to address(as address)**

AIM element: address  
 Source: ISO 10303-41  
 Reference Path: organization <-  
 organizational\_address.organizations[i]  
 organizational\_address <=  
 address

**5.1.13.5 Person**

AIM element: person  
 Source: ISO 10303-41

**5.1.13.5.1 person\_id**

AIM element: person.id  
 Source: ISO 10303-41

**5.1.13.5.2 person\_name**

AIM element: [person.first\_name]  
 [person.last\_name]  
 Source: ISO 10303-41

**5.1.13.5.3 person\_phone\_number**

AIM element: address.telephone\_number  
 Source: ISO 10303-41  
 Reference Path: person <-  
 personal\_address.people[i]  
 personal\_address <=  
 address  
 address.telephone\_number

**5.1.13.5.4 title**

AIM element: person.prefix\_titles  
 Source: ISO 10303-41

### 5.1.13.5 person to address (as address)

AIM element: address  
Source: ISO 10303-41  
Reference Path: person <-  
personal\_address.people[i]  
personal\_address <=  
address

### 5.1.13.6 Person\_in\_organization

AIM element: person\_and\_organization  
Source: ISO 10303-41

#### 5.1.13.6.1 role

AIM element: person\_and\_organization\_role.name  
Source: ISO 10303-41  
Reference Path: person\_and\_organization <-  
person\_and\_organization\_assignment.assigned\_person\_and\_organization  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.role ->  
person\_and\_organization\_role  
person\_and\_organization\_role.name

#### 5.1.13.6.2 person\_in\_organization to organization (as company)

AIM element: PATH  
Reference Path: person\_and\_organization  
person\_and\_organization.the\_organization ->  
organization

#### 5.1.13.6.3 person\_in\_organization to person (as employee)

AIM element: PATH  
Reference Path: person\_and\_organization  
person\_and\_organization.the\_person ->  
person

### 5.1.13.7 Status\_authority

AIM element: approval  
Source: ISO 10303-41  
Rule: ,

### 5.1.13.7.1 approval\_title

AIM element: approval\_role.role  
 Source: ISO 10303-41  
 Reference path: approval <-  
 approval\_person\_organization.authorized\_approval  
 approval\_person\_organization  
 approval\_person\_organization.role ->  
 approval\_role  
 approval\_role.role

### 5.1.13.7.2 date

AIM element: calendar\_date  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference path: approval <-  
 approval\_date\_time.dated\_approval  
 approval\_date\_time  
 approval\_date\_time.date\_time ->  
 date\_time\_select  
 date\_time\_select = date  
 date =>  
 calendar\_date

### 5.1.13.7.3 status\_authority\_to\_planning\_group\_member (as is\_given\_by)

AIM element: PATH  
 Reference path: approval<-  
 approval\_person\_organization.authorized\_approval  
 approval\_person\_organization  
 approval\_person\_organization.person\_organization->  
 person\_organization\_select  
 person\_organization\_select = person  
 person

## 5.1.14 Part\_model UoF

### 5.1.14.1 Cast\_part

AIM element: cast\_part  
 Source: ISO 10303-223  
 Reference Path: cast\_part<=  
 product\_definition

### 5.1.14.1.1 **cast\_part to casting\_design\_feature (as cast\_feature)**

AIM element: PATH  
Reference Path: cast\_part<=  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
shape\_aspect=>  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
casting\_design\_feature

### 5.1.14.2 **Cast\_part\_with\_rigging**

AIM element: cast\_part\_with\_rigging  
Source: ISO 10303-223  
Reference Path: cast\_part\_with\_rigging<=  
product\_definition

### 5.1.14.2.1 **cast\_part\_with\_rigging to casting\_design\_feature (as cast\_feature)**

AIM element: PATH  
Reference Path: cast\_part\_with\_rigging<=  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <=  
property\_definition.definition  
property\_definition =>  
product\_definition\_shape <=  
shape\_aspect.of\_shape  
shape\_aspect=>  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
casting\_design\_feature



### 5.1.14.2.2 cast\_part\_with\_rigging to gating\_system (as rigging)

AIM element: PATH  
 Reference Path: cast\_part\_with\_rigging<=  
                   product\_definition  
                   characterized\_product\_definition = product\_definition  
                   characterized\_product\_definition  
                   characterized\_definition = characterized\_product\_definition  
                   characterized\_definition <=  
                   property\_definition.definition  
                   property\_definition =>  
                   product\_definition\_shape <=  
                   shape\_aspect.of\_shape  
                   shape\_aspect=>  
                   casting\_instanced\_feature<=  
                   casting\_feature\_definition=>  
                   gating\_design\_feature=>  
                   gating\_system

### 5.1.14.3 Design\_part

AIM element: design\_part  
 Source: ISO 10303-223  
 Reference Path: design\_part  
                   product\_definition

#### 5.1.14.3.1 design\_part to Manufacturing\_design\_feature (as design\_feature)

AIM element: PATH  
 Reference Path: design\_part  
                   product\_definition  
                   characterized\_product\_definition = product\_definition  
                   characterized\_product\_definition  
                   characterized\_definition = characterized\_product\_definition  
                   characterized\_definition <=  
                   property\_definition.definition  
                   property\_definition =>  
                   product\_definition\_shape <=  
                   shape\_aspect.of\_shape  
                   shape\_aspect=>  
                   instanced\_feature<=  
                   feature\_definition

#### 5.1.14.4 Master\_sample

AIM element: master\_sample  
Source: ISO 10303-223  
Reference Path: master\_sample<=  
cast\_part<=  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
{application\_context\_element  
application\_context\_element.name='part definition'}  
product\_definition\_context.life\_cycle\_stage='finish casting'

##### 5.1.14.4.1 master\_sample to date (as production\_date)

AIM element: calendar\_date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: master\_sample<=  
cast\_part<=  
product\_definition  
date\_item = product\_definition  
date\_item <-  
applied\_date\_assignment.items [i]  
applied\_date\_assignment <=  
date\_assignment  
{date\_assignment.role ->  
date\_role  
date\_role.name='production date'}  
date\_assignment.assigned\_date ->  
date =>  
calendar\_date

##### 5.1.14.4.2 master\_sample to design\_part (as product\_specification\_used)

AIM element: PATH  
Reference Path: master\_sample<=  
cast\_part<=  
product\_definition  
product\_definition\_relationship.relying\_product\_definition  
product\_definition\_relationship  
product\_definition\_relationship.related\_product\_definition->  
product\_definition=>  
design\_part

**5.1.14.4.3 master\_sample to heat (as belongs\_to\_heat)**

AIM element: effectivity  
 Source: ISO 10303-41  
 Reference Path: master\_sample<=  
                   cast\_part<=  
                   product\_definition  
                   effectivity\_item=product\_definition<-  
                   applied\_effectivity\_assignment.items[i]  
                   applied\_effectivity\_assignment<=  
                   effectivity\_assignment  
                   effectivity\_assignment.assigned\_effectivity->  
                   effectivity  
                   dated\_effectivity

**5.1.14.4.4 master\_sample to lot (as belongs\_to\_lot)**

AIM element: effectivity  
 Source: ISO 10303-41  
 Reference Path: master\_sample<=  
                   cast\_part<=  
                   product\_definition  
                   effectivity\_item=product\_definition<-  
                   applied\_effectivity\_assignment.items[i]  
                   applied\_effectivity\_assignment<=  
                   effectivity\_assignment  
                   effectivity\_assignment.assigned\_effectivity->  
                   effectivity  
                   dated\_effectivity

**5.1.14.4.5 master\_sample to process\_plan (as process\_plan\_used)**

AIM element: PATH  
 Reference Path: master\_sample<=  
                   cast\_part<=  
                   product\_definition  
                   characterized\_product\_definition = product\_definition  
                   characterized\_product\_definition <-  
                   process\_product\_association.defined\_product  
                   process\_product\_association  
                   process\_product\_association.description='process plan used'  
                   process\_product\_association.process ->  
                   product\_definition\_process =>  
                   process\_plan\_version

#### 5.1.14.4.6 master\_sample to external\_schema\_definition (as process\_plan\_used\_by)

AIM element: PATH  
Reference Path: master\_sample<=  
cast\_part<=  
product\_definition  
external\_identification\_item=product\_definition  
external\_identification\_item<-  
applied\_external\_identification\_assignment.items[i]<=  
external\_identification\_assignment  
external\_identification\_assignment.source->  
external\_source<-  
external\_defined\_item=>  
externally\_defined\_schema

#### 5.1.14.5 Part\_relationship

AIM element: product\_definition\_usage  
Source: ISO 10303-223  
Reference Path: product\_definition\_usage<=  
product\_definition\_relationship

##### 5.1.14.5.1 Part\_relationship to design\_part (as as\_designed)

AIM element: PATH  
Reference Path: product\_definition\_usage<=  
product\_definition\_relationship  
(product\_definition\_relationship.name= 'as designed to be cast')  
product\_definition\_relationship.related\_product\_definition->  
product\_definition=>  
design\_part

##### 5.1.14.5.2 Part\_relationship to cast\_part (as to\_be\_cast)

AIM element: PATH  
Reference Path: product\_definition\_usage<=  
product\_definition\_relationship  
(product\_definition\_relationship.name= 'as cast')  
product\_definition\_relationship.relatng\_product\_definition->  
product\_definition=>  
cast\_part

### 5.1.14.5.3 Part\_relationship to tooling (as tooling\_equipment)

AIM element: PATH  
 Reference Path: product\_definition\_usage<=  
 product\_definition\_relationship  
 {product\_definition\_relationship.related\_product\_definition->  
 product\_definition<-  
 {product\_definition=>  
 design\_part}  
 product\_definition\_relationship.relateing\_product\_definition  
 product\_definition\_relationship=>  
 product\_definition\_usage  
 (product\_definition\_relationship.name= 'tooling')  
 product\_definition\_relationship.related\_product\_definition  
 product\_definition=>  
 casting\_product\_definition

### 5.1.14.5.4 Part\_relationship to cast\_part\_with\_rigging (as to\_be\_cast)

AIM element: PATH  
 Reference Path: product\_definition\_usage<=  
 product\_definition\_relationship  
 (product\_definition\_relationship.name= 'as cast')  
 product\_definition\_relationship.relateing\_product\_definition->  
 product\_definition

### 5.1.14.6 Part\_version

AIM element: product\_definition\_formation  
 Source: ISO 10303-41  
 Reference Path: product\_definition\_formation<-  
 product\_definition.formation  
 product\_definition  
 product\_definition.frame\_of\_reference->  
 product\_definition\_context<=  
 application\_context\_element  
 application\_context\_element.name='part definition'

#### 5.1.14.6.1 part\_description

AIM element: product.description  
 Source: ISO 10303-41  
 Reference Path: product\_definition\_formation  
 product\_definition\_formation.of\_product ->  
 product  
 product.description

ISO 10303-223:2008 (E)

### 5.1.14.6.2 part\_id

AIM element: product.id  
Source: ISO 10303-41  
Reference Path: product\_definition\_formation  
product\_definition\_formation.of\_product ->  
product  
product.id

### 5.1.14.6.3 part\_name

AIM element: product.name  
Source: ISO 10303-41  
Reference Path: product\_definition\_formation  
product\_definition\_formation.of\_product ->  
product  
product.name

### 5.1.14.6.4 part\_revision\_id

AIM element: product\_definition\_formation.id  
Source: ISO 10303-41  
Rules: product\_requires\_version – (See 5.2.4.22)

### 5.1.14.6.5 security\_classification

AIM element: security\_classification  
Source: ISO 10303-41  
Rules: part\_requires\_security\_classification – (See 5.2.4.23)  
Reference Path: product\_definition\_formation  
security\_classification\_item = product\_definition\_formation  
security\_classification\_item<-  
applied\_security\_classification\_assignment.items[i]  
applied\_security\_classification\_assignment<=  
security\_classification\_assignment  
security\_classification\_assignment.assigned\_security\_classification ->  
security\_classification

**5.1.14.6.6 part\_version to approval (as manufacture\_authorization)**

AIM element: PATH  
 Rules: part\_to\_approval – (See 5.2.4.21)  
 Reference Path: product\_definition\_formation  
 approved\_item = product\_definition\_formation  
 approved\_item <-  
 applied\_approval\_assignment.items[i]  
 applied\_approval\_assignment <=  
 approval\_assignment  
 approval\_assignment.assigned\_approval ->  
 approval

**5.1.14.6.7 part\_version to customer\_casting\_requirement (as customer\_requirement)**

AIM element: PATH  
 Reference Path: product\_definition\_formation <-  
 product\_definition.formation  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition=>  
 customer\_casting\_requirement

### 5.1.14.6.8 part\_version to material (as material\_definition)

AIM element: PATH  
Rules: material\_is\_specified\_for\_part – (See 5.2.4.19)  
Reference Path: product\_definition\_formation <-  
product\_definition.formation  
product\_definition <-  
product\_definition\_relationship.relateing\_product\_definition  
{product\_definition\_relationship.description='raw material'  
product\_definition\_relationship =>  
product\_definition\_usage =>  
make\_from\_usage\_option}  
product\_definition\_relationship  
product\_definition\_relationship.related\_product\_definition ->  
product\_definition  
{product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
material\_designation.definitions[i]  
material\_designation}

### 5.1.14.6.9 part\_version to design\_reference (as documented\_by)

AIM element: PATH  
Reference Path: product\_definition\_formation  
document\_reference\_item=product\_definition\_formation  
document\_reference\_item<-  
applied\_document\_reference.items [i]  
applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document =>  
design\_reference

### 5.1.14.6.10 part\_version to ordered\_part (as quantity\_ordered)

AIM element: PATH  
Reference Path: product\_definition\_formation  
applied\_ordered\_item = product\_definition\_formation  
applied\_ordered\_item <-  
ordered\_part.items[i]  
ordered\_part



**5.1.14.6.11 part\_version to organization (as manufactured\_by)**

AIM element: PATH  
 Reference Path: product\_definition\_formation  
                   organization\_item = product\_definition\_formation  
                   organization\_item <-  
                   applied\_organization\_assignment.items[i]  
                   applied\_organization\_assignment <=  
                   {organization\_assignment  
                   organization\_assignment.role ->  
                   organization\_role  
                   organization\_role.name = 'manufacturer'}  
                   organization\_assignment  
                   organization\_assignment.assigned\_organization ->  
                   organization

**5.1.14.6.12 part\_version to organization (as owned\_by)**

AIM element: PATH  
 Reference Path: product\_definition\_formation  
                   organization\_item = product\_definition\_formation  
                   organization\_item <-  
                   applied\_organization\_assignment.items[i]  
                   applied\_organization\_assignment <=  
                   {organization\_assignment  
                   organization\_assignment.role ->  
                   organization\_role  
                   organization\_role.name = 'owner'}  
                   organization\_assignment  
                   organization\_assignment.assigned\_organization ->  
                   organization

### 5.1.14.6.13 part\_version to person\_in\_organization (as manufactured\_by)

AIM element: PATH  
Reference Path: product\_definition\_formation  
person\_and\_organization\_item = product\_definition\_formation  
person\_and\_organization\_item <-  
applied\_and\_organization\_assignment.items[i]  
applied\_person\_and\_organization\_assignment <=  
{person\_and\_organization\_assignment  
person\_and\_organization\_assignment.role ->  
person\_and\_organization\_role  
person\_and\_organization\_role.name = 'manufacturer'}  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.assigned\_person\_and\_organization ->  
person\_and\_organization

### 5.1.14.6.14 part\_version to person\_in\_organization (as owned\_by)

AIM element: PATH  
Reference Path: product\_definition\_formation  
person\_and\_organization\_item = product\_definition\_formation  
person\_and\_organization\_item <-  
applied\_person\_and\_organization\_assignment.items[i]  
applied\_person\_and\_organization\_assignment <=  
{person\_and\_organization\_assignment  
person\_and\_organization\_assignment.role ->  
person\_and\_organization\_role  
person\_and\_organization\_role.name = 'owner'}  
person\_and\_organization\_assignment  
person\_and\_organization\_assignment.assigned\_person\_and\_organization ->  
person\_and\_organization

### 5.1.14.6.15 part\_version to process\_plan\_version (as casting\_defined\_by)

AIM element: PATH  
Reference path: product\_definition\_formation <-  
product\_definition.formation  
product\_definition  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition <-  
process\_product\_association.defined\_product  
process\_product\_association  
process\_product\_association.process ->  
product\_definition\_process =>  
process\_plan\_version

**5.1.14.6.16 part\_version to process\_plan\_version (as casting\_defined\_by)**

AIM element: PATH  
 Reference Path: product\_definition\_formation <-  
 external\_identification\_item=product\_definition\_formation  
 external\_identification\_item<-  
 applied\_external\_identification\_assignment<=  
 external\_identification\_assignment  
 external\_identification\_assignment.source->  
 external\_source<-  
 external\_defined\_item=>  
 externally\_defined\_schema

**5.1.14.6.17 part\_version to property (as property\_characteristics)**

AIM element: PATH  
 Reference Path: product\_definition\_formation <-  
 product\_definition.formation  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition

**5.1.14.6.18 part\_version to shape (as physical\_form)**

AIM element: PATH  
 Reference Path: product\_definition\_formation <-  
 product\_definition.formation  
 product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition =>  
 product\_definition\_shape

**5.1.14.7 Red\_lined\_part**

AIM element: red\_lined\_part  
 Source: ISO 10303-223  
 Reference Path: red\_lined\_part<=  
 product\_definition

### 5.1.14.7.1 red\_lined\_part to feature (as mark\_up\_feature)

AIM element: PATH  
Reference Path: red\_lined\_part<=  
product\_definition <-  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition<=  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'red lined part reference occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'feature reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect =>  
(casting\_instanced\_feature)  
(instanced\_feature)

### 5.1.14.7.2 red\_lined\_part to material (as mark\_up\_material)

AIM element: PATH  
 Reference Path: red\_lined\_part<=  
 product\_definition <-  
 product\_definition\_relationship.relatng\_product\_definition  
 {product\_definition\_relationship.description='mark up material'  
 product\_definition\_relationship =>  
 product\_definition\_usage =>  
 make\_from\_usage\_option}  
 product\_definition\_relationship  
 product\_definition\_relationship.related\_product\_definition ->  
 product\_definition  
 {product\_definition  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 material\_designation.definitions[i]  
 material\_designation}

### 5.1.14.7.3 red\_lined\_part to property (as mark\_up\_property)

AIM element: PATH  
 Reference Path: red\_lined\_part<=  
 product\_definition <-  
 characterized\_product\_definition = product\_definition  
 characterized\_product\_definition  
 characterized\_definition = characterized\_product\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition  
 {id\_attribute\_select=property\_definition  
 id\_attribute\_select<-  
 id\_attribute.identified\_item  
 id\_attribute  
 id\_attribute.attribute\_value='mark up property'}

#### 5.1.14.7.4 red\_lined\_part to shape\_aspect (as mark\_up\_shape)

AIM element: PATH  
Reference Path: red\_lined\_part<=  
product\_definition <-  
characterized\_product\_definition = product\_definition  
characterized\_product\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_definition <-  
property\_definition.definition  
property\_definition<=  
product\_definition\_shape <-  
shape\_aspect.of\_shape  
{shape\_aspect  
shape\_aspect.description = 'red lined part reference occurrence'}  
shape\_aspect <-  
shape\_aspect\_relationship.related\_shape\_aspect  
{[shape\_aspect\_relationship =>  
shape\_defining\_relationship]  
[shape\_aspect\_relationship  
shape\_aspect\_relationship.description = 'shape aspect reference usage']}  
shape\_aspect\_relationship  
shape\_aspect\_relationship.relating\_shape\_aspect ->  
shape\_aspect

#### 5.1.14.8 Tooling

AIM element: casting\_product\_definition  
Source: ISO 10303-223  
Reference Path: casting\_product\_definition<=  
product\_definition  
product\_definition.frame\_of\_reference->  
product\_definition\_context<=  
application\_context\_element  
application\_context\_element.name='functional definition'

### 5.1.14.8.1 tooling to cast\_part (as tooling designed\_for)

AIM element: PATH  
 Reference Path: casting\_product\_definition<=  
 product\_definition<-  
 product\_definition\_relationship.related\_product\_definition  
 product\_definition\_relationship  
 {product\_definition\_relationship=>  
 Product\_definitoin\_usage}  
 product\_definition\_relationship.relatig\_product\_definition->  
 product\_definition=>  
 cast\_part

### 5.1.14.8.2 toolong to date ( as date\_last\_used)

AIM element: PATH  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference Path: casting\_product\_definition<=  
 product\_definition  
 date\_item=product\_definition  
 date\_item<-  
 applied\_date\_assignment.items[i]  
 applied\_date\_assignment  
 applied\_date\_assignment<=  
 date\_assignment  
 date\_assignment.assigned\_date->  
 date

## 5.1.15 Process\_activities UoF

### 5.1.15.1 Activity

AIM element: process\_plan\_activity  
 Source: ISO 10303-223  
 Reference Path: process\_plan\_activity<=  
 action\_method

### 5.1.15.1.1 activity\_number

AIM element: count\_measure  
Source: ISO 10303-41  
Reference Path: process\_plan\_activity<=  
action\_method  
action\_method=characterized\_action\_definition  
characterized\_action\_definition <=  
action\_property.definition  
action\_property <=  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation ->  
representation  
representation.items [i]->  
representation\_item =>  
{representation\_item.name ='activity number'}  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.value\_component->  
measure\_value  
measure\_value=count\_measure  
count\_measure

### 5.1.15.1.2 description

AIM element: action\_method.description  
Source: ISO 10303-41  
Reference Path: process\_plan\_activity<=  
action\_method  
action\_method.description

### 5.1.15.1.3 frequency

AIM element: action\_resource.description  
Source: ISO 10303-41  
Reference Path: process\_plan\_activity<=  
action\_method  
supported\_item=action\_method  
supported\_item <=  
action\_resource.usage[i]  
action\_resource  
{action\_resource.name='frequency'}  
action\_resource.description



**5.1.15.1.4 name**

AIM element: action\_method.name  
 Source: ISO 10303-41  
 Reference Path: process\_plan\_activity<=  
                   action\_method  
                   action\_method.name

**5.1.15.1.5 activity to activity\_execution\_result to activity (as execution\_result)**

AIM element: PATH  
 Reference Path: process\_plan\_activity<=  
                   action\_method<-  
                   action.chosen\_method  
                   action=>  
                   executed\_action=>  
                   activity\_execution\_result

**5.1.15.1.6 activity to organization (as performed\_by\_organization\_id)**

AIM element: organization.id  
 Source: ISO 10303-41  
 Reference Path: process\_plan\_activity  
                   organization\_item=process\_plan\_activity  
                   organization\_item <-  
                   applied\_organization\_assignment.items [i]  
                   applied\_organization\_assignment <=  
                   organization\_assignment  
                   {organization\_assignment.role->  
                   organization\_role  
                   organization\_role.name='activity organization id'}  
                   organization\_assignment.assigned\_organization ->  
                   organization  
                   organization.id

### 5.1.15.1.7 activity to special\_instruction (as activity\_information)

AIM element: PATH  
Reference Path: process\_plan\_activity<=  
action\_method  
action\_method=characterized\_action\_definition  
characterized\_action\_definition <-  
action\_property.definition  
action\_property <-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation ->  
representation  
representation.items [i]->  
representation\_item =>  
(representation\_item.name='special instruction')  
descriptive\_representation\_item

### 5.1.15.1.8 activity to supplemental\_document (as process\_specification)

AIM element: PATH  
Reference Path: process\_plan\_activity<=  
action\_method=>  
action\_method\_with\_associated\_documents  
action\_method\_with\_associated\_documents.documents[i]->  
document=>  
process\_plan\_specification  
document.kind->  
document\_type  
document\_type.product\_data\_type='supplemental document'

### 5.1.15.1.9 activity to performance\_rate (as duration)

AIM element: PATH  
Reference Path: process\_plan\_activity<=  
action\_method  
action\_method=characterized\_action\_definition  
characterized\_action\_definition <-  
action\_property.definition  
action\_property=>  
(allowed\_time)  
(production\_rate)

**5.1.15.1.10 activity to design\_reference (as graphics\_representation)**

AIM element: PATH  
 Reference Path: process\_plan\_activity<=  
                   action\_method=>  
                   action\_method\_with\_associated\_documents  
                   action\_method\_with\_associated\_documents.documents[i]->  
                   document=>  
                   design\_reference

**5.1.15.1.11 activity to illustration (as graphics\_representation)**

AIM element: PATH  
 Reference Path: process\_plan\_activity<=  
                   action\_method=>  
                   action\_method\_with\_associated\_documents  
                   action\_method\_with\_associated\_documents.documents[i]->  
                   document=>  
                   illustration

**5.1.15.1.12 activity to view\_reference (as graphics\_representation)**

AIM element: PATH  
 Reference Path: process\_plan\_activity<=  
                   action\_method=>  
                   action\_method\_with\_associated\_documents  
                   action\_method\_with\_associated\_documents.documents[i]->  
                   document=>  
                   view\_reference

**5.1.15.1.13 activity to cast\_part (as output\_item)**

AIM element: PATH  
 Reference Path: process\_plan\_activity<=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action\_method  
                   action\_method\_assignment =>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i] ->  
                   action\_item  
                   action\_item=product\_definition  
                   product\_definition=>  
                   cast\_part

#### **5.1.15.1.14 activity to cast\_part\_with\_rigging (as output\_item)**

AIM element: PATH  
Reference Path: process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment =>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i] ->  
action\_item  
action\_item=product\_definition  
product\_definition=>  
cast\_part\_with\_rigging

#### **5.1.15.1.15 activity to generic\_manufacturing\_resource (as consumable\_resource)**

AIM element: PATH  
Reference Path: process\_plan\_activity<=  
action\_method  
supported\_item=action\_method  
supported\_item <-  
action\_resource.usage  
action\_resource=>  
generic\_manufacturing\_resource

#### **5.1.15.2 Allowed\_time**

AIM element: allowed\_time  
Source: ISO 10303-223  
Reference Path: allowed\_time <=  
action\_property

### 5.1.15.2.1 allowance\_factor

AIM element: time\_measure\_with\_unit  
 Source: ISO 10303-41  
 Reference Path: allowed\_time <=  
 action\_property <-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation ->  
 representation  
 representation.items [i] ->  
 representation\_item =>  
 {representation\_item.name ='allowance\_factor'}  
 measure\_representation\_item <=  
 measure\_with\_unit =>  
 count\_measureset

### 5.1.15.2.2 allowed\_type

AIM element: descriptive\_representation\_item  
 Source: ISO 10303-41  
 Reference Path: allowed\_time <=  
 action\_property <-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation->  
 representation  
 representation.items[i]->  
 representation\_item=>  
 {representation\_item.name='allowed type'}  
 descriptive\_representation\_item

### 5.1.15.2.3 standard\_time

AIM element: time\_measure\_with\_unit  
Source: ISO 10303-41  
Reference Path: allowed\_time <=  
action\_property <-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation ->  
representation  
representation.items [i] ->  
representation\_item =>  
{representation\_item.name ='standard\_time'}  
measure\_representation\_item <=  
measure\_with\_unit =>  
time\_measure\_with\_unit

### 5.1.15.3 Alternate\_activity

AIM element: process\_plan\_activity  
Source: ISO 10303-223  
Reference Path: process\_plan\_activity<=  
action\_method

#### 5.1.15.3.1 primary\_activity

AIM element: process\_plan\_activity  
Source: ISO 10303-223  
Reference Path: process\_plan\_activity<=  
action\_method<-  
action\_method\_relationship.related\_method  
action\_method\_relationship=>  
alternate\_action\_method\_relationship  
action\_method\_relationship  
action\_method\_relationship.relying\_method ->  
action\_method=>  
process\_plan\_activity

### 5.1.15.4 Ancillary\_activity

AIM element: ancillary\_activity  
Source: ISO 10303-223  
Reference Path: ancillary\_activity <=  
process\_plan\_activity<=  
action\_method

**5.1.15.4.1 method**

AIM element: action\_method.purpose  
 Source: ISO 10303-41  
 Reference Path: ancillary\_activity <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method  
 action\_method.purpose

**5.1.15.5 Ancillary\_setup**

AIM element: ancillary\_setup  
 Source: ISO 10303-223  
 Reference Path: ancillary\_setup <=  
 process\_plan\_activity<=  
 action\_method

**5.1.15.5.1 setup\_type**

AIM element: action\_method.purpose  
 Source: ISO 10303-41  
 Reference Path: ancillary\_setup <=  
 process\_plan\_activity<=  
 action\_method  
 action\_method.purpose

**5.1.15.6 Assembly**

AIM element: assembly  
 Source: ISO 10303-223  
 Reference Path: assembly<=  
 non\_casting\_operation<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

### 5.1.15.6.1 foam\_pattern\_coating\_density

AIM element: PATH  
Reference Path: assembly<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='foam pattern coating density'  
descriptivre\_representation\_item

### 5.1.15.6.2 pattern\_section\_assembly\_sequence

AIM element: PATH  
Reference Path: assembly<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='pattern section assembly sequence'  
descriptivre\_representation\_item



### 5.1.15.7 Bench\_inspection

AIM element: bench\_inspection  
 Source: ISO 10303-223  
 Reference Path: bench\_inspection<=  
 inspection\_activity <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

### 5.1.15.8 Casting\_activity

AIM element: casting\_activity  
 Source: ISO 10303-223  
 Reference Path: casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

#### 5.1.15.8.1 casting\_activity to external\_schema\_definition (as micro\_plan\_-reference)

AIM element: PATH  
 Reference Path: casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=externally\_defined\_item  
 externally\_defined\_item=>  
 externally\_defined\_schema

### 5.1.15.9 General\_foundry\_process

AIM element: casting\_process  
Source: ISO 10303-223  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

#### 5.1.15.9.1 core\_making\_process

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'shell')  
(descriptive\_representation\_item.description = 'no bake')  
(descriptive\_representation\_item.description = 'lost form')  
(descriptive\_representation\_item.description = 'gas cured')  
(descriptive\_representation\_item.description = 'green sand')}

### 5.1.15.9.2 type\_of\_operation

AIM element: action\_method.name  
 Source: ISO 10303-41  
 Reference Path: casting\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method  
 action\_method.name  
 {(action\_method.name='type of operation')  
 (action\_method.name='core setting')  
 (action\_method.name='die assembly')  
 (action\_method.name='die handling')  
 (action\_method.name='die temperature control')  
 (action\_method.name='furnace loading')  
 (action\_method.name='gate system removal')  
 (action\_method.name='grinding')  
 (action\_method.name='inspection')  
 (action\_method.name='ladle loading')  
 (action\_method.name='loading injection system')  
 (action\_method.name='melt raw material')  
 (action\_method.name='mould closing')  
 (action\_method.name='mould making process')  
 (action\_method.name='package and shipping')  
 (action\_method.name='part routing')  
 (action\_method.name='pattern handling')  
 (action\_method.name='pouring process')  
 (action\_method.name='refractor removal')  
 (action\_method.name='removal from die')  
 (action\_method.name='shakeout')  
 (action\_method.name='slurry process')  
 (action\_method.name='sand removal')  
 (action\_method.name='tool handling')  
 (action\_method.name='wax removal') }

### 5.1.15.9.3 general\_foundry\_process to equipment\_setup (as identifies)

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_relationship.related\_method  
action\_method\_relationship=>  
defining\_action\_method\_relationship  
action\_method\_relationship  
action\_method\_relationship.relating\_method ->  
action\_method=>  
machine\_setup=>  
casting\_equipment\_setup

### 5.1.15.9.4 general\_foundry\_process to mismatch\_tolerance (as tolerances)

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
mismatch\_tolerance

### 5.1.15.9.5 general\_foundry\_process to production\_core\_box (as core\_to\_be\_made)

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->

```

action_method_item
action_method_item=product_definition
product_definition=>
production_core_box

```

### 5.1.15.9.6 general\_foundry\_process to numeric\_parameter (as process\_parameter)

```

AIM element:    PATH
Reference Path: casting_process<=
                casting_activity<=
                manufacturing_activity<=
                process_plan_activity<=
                action_method
                characterized_action_definition=action_method
                characterized_action_definition<-
                action_property.definition
                action_property<-
                action_property_representation.property
                action_property_representation
                action_property_representation.representation
                representation
                representation.items[i] ->
                representation_item
                representation_item =>
                measure_representation_item
                {measure_representation_item <=
                measure_with_unit =>
                length_measure_with_unit}

```

### 5.1.15.9.7 **general\_foundry\_process to property\_relationship (as process\_parameter)**

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
property\_relationship

### 5.1.15.9.8 **general\_foundry\_process to shape\_aspect (as described\_by)**

AIM element: PATH  
Reference Path: casting\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=shape\_aspect  
shape\_aspect

### 5.1.15.10 **Customer\_simulation**

AIM element: simulation\_process  
Source: ISO 10303-223  
Reference Path: customer\_simulation<=  
simulation\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

### 5.1.15.10.1 customer\_simulations to customer\_tool\_design\_report (as defect\_results)

AIM element: PATH  
 Reference Path: customer\_simulation<=  
 simulation\_process <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method=>  
 action\_method\_with\_associated\_documents  
 action\_method\_with\_associated\_documents.documents[i]->  
 document=>  
 document.kind->  
 document\_type  
 document\_type.product\_data\_type='customer tool design report'

### 5.1.15.10.2 customer\_simulation to customer\_design\_deficiency\_report (as suggestion\_for\_changes)

AIM element: PATH  
 Reference Path: customer\_simulation<=  
 simulation\_process <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method=>  
 action\_method\_with\_associated\_documents  
 action\_method\_with\_associated\_documents.documents[i]->  
 document=>  
 document.kind->  
 document\_type  
 document\_type.product\_data\_type='customer design deficiency report'

### 5.1.15.11 Dimensional\_measurement\_inspection

AIM element: dimensional\_measurement\_inspection  
 Source: ISO 10303-223  
 Reference Path: dimensional\_measurement\_inspection<=  
 inspection\_activity <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

### 5.1.15.11.1 dimensional\_measurement\_inspection to external\_schema\_definition (as iso10303\_219\_data)

AIM element: PATH  
Reference Path: dimensional\_measurement\_inspection<=  
inspection\_activity <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=externally\_defined\_item  
externally\_defined\_item=>  
externally\_defined\_schema

### 5.1.15.12 Equipment\_process

AIM element: casting\_equipment\_process  
Source: ISO 10303-223  
Reference path: casting\_equipment\_process<=  
machining\_process=>  
manufacturing\_process <=  
action\_method

### 5.1.15.12.1 equipment\_process to equipment (as required\_equipment)

AIM element: PATH  
Reference path: casting\_equipment\_process<=  
machining\_process=>  
manufacturing\_process <=  
action\_method  
supported\_item=action\_method  
supported\_item <-  
action\_resource.usage[i]  
action\_resource =>  
machine=>  
casting\_equipment



### 5.1.15.13 Equipment\_setup

AIM element: casting\_equipment\_setup  
 Source: ISO 10303-223  
 Reference Path: casting\_equipment\_setup <=  
 machine\_setup<=  
 process\_plan\_activity<=  
 action\_method

#### 5.1.15.13.1 setup\_instruction

AIM element: action\_method.description  
 Source: ISO 10303-41  
 Reference Path: casting\_equipment\_setup <=  
 machine\_setup<=  
 process\_plan\_activity<=  
 action\_method  
 action\_method.description

#### 5.1.15.13.2 equipment\_setup to equipment (as identifies)

AIM element: PATH  
 Reference Path: casting\_equipment\_setup <=  
 machine\_setup<=  
 process\_plan\_activity<=  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_resource\_requirement.operations[i]  
 action\_resource\_requirement=>  
 requirement\_for\_action\_resource  
 requirement\_for\_action\_resource.resources [i]->  
 action\_resource=>  
 machine=>  
 casting\_equipment

### 5.1.15.14 Fabrication

AIM element: fabrication  
 Source: ISO 10303-223  
 Reference Path: fabrication<=  
 non\_casting\_operation<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

### 5.1.15.15 Heat\_treat

AIM element: heat\_treat  
Source: ISO 10303-223  
Reference Path: heat\_treat<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

#### 5.1.15.15.1 Heat\_treat to heat\_treat\_requirement (as requirement)

AIM element: PATH  
Reference Path: heat\_treat<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
customer\_casting\_requirement=>  
process\_requirement=>  
heat\_treat\_requirement

### 5.1.15.16 Impregnation\_activity

AIM element: impregnation\_activity  
Source: ISO 10303-223  
Reference Path: impregnation\_activity <=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

### 5.1.15.17 Inspection\_activity

AIM element: inspection\_activity  
 Source: ISO 10303-223  
 Reference Path: inspection\_activity <=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method

#### 5.1.15.17.1 inspection\_level

AIM element: PATH  
 Reference Path: heat\_treat<=  
                   non\_casting\_operation<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action\_method  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definition  
                   property\_definition  
                   represented\_definition=property\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation.used\_representation->  
                   representation  
                   representation.items[i] ->  
                   representation\_item  
                   representation\_item.name='inspection level'  
                   measure\_representation\_item <=  
                   measure\_with\_unit  
                   measure\_with\_unit.value\_component->  
                   measure\_value  
                   measure\_value=count\_measure  
                   count\_measure

### 5.1.15.17.2 reason\_for\_inspection

AIM element: PATH  
Reference Path: heat\_treat<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='inspection reason'  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description='initial inspection')  
(descriptive\_representation\_item.description='engineering change')  
(descriptive\_representation\_item.description='tooling replacement')  
(descriptive\_representation\_item.description='tooling transfer')  
(descriptive\_representation\_item.description='tooling refurbishment')  
(descriptive\_representation\_item.description='discrepancy correction')  
(descriptive\_representation\_item.description='material or supplier source change')  
(descriptive\_representation\_item.description='production sample set')}

### 5.1.15.17.3 type\_of\_inspection

AIM element: PATH  
 Reference Path: heat\_treat<=  
 non\_casting\_operation<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definition  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='inspection type'  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description='destructive')  
 (descriptive\_representation\_item.description='dimensional')  
 (descriptive\_representation\_item.description='functional testing')  
 (descriptive\_representation\_item.description='gauging')  
 (descriptive\_representation\_item.description='hardness property')  
 (descriptive\_representation\_item.description='internal property')  
 (descriptive\_representation\_item.description='leak testing')  
 (descriptive\_representation\_item.description='material composition')  
 (descriptive\_representation\_item.description='surface')  
 (descriptive\_representation\_item.description='visual')}

#### 5.1.15.17.4 inspection\_activity to approval (as foundry approval)

AIM element: PATH  
Reference Path: heat\_treat<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
approval\_item = action\_method  
approval\_item <-  
applied\_approval\_assignment.items[i]  
applied\_approval\_assignment <=  
approval\_assignment  
role\_select=approval\_assignment<-  
{role\_association.item\_with\_role  
role\_assignment  
role\_assignment.role->  
object\_role  
object\_role.name='foundry approval'}  
approval\_assignment.assigned\_approval ->  
approval

#### 5.1.15.17.5 inspection\_activity to casting\_verification (as quality\_assurance)

AIM element: PATH  
Reference Path: heat\_treat<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_request\_solution.method  
action\_request\_solution  
action\_request\_solution.request->  
versioned\_action\_request=>  
casting\_verification

**5.1.15.17.6 inspection\_activity to part\_version (as inspected\_part)**

AIM element: PATH  
 Reference Path: heat\_treat<=  
 non\_casting\_operation<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definition  
 product\_definition  
 product\_definition.formation->  
 product\_definition\_formation

**5.1.15.17.7 inspection\_activity to product\_quality\_report (as inspection\_report)**

AIM element: PATH  
 Reference Path: inspection\_activity <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method=>  
 action\_method\_with\_associated\_documents  
 action\_method\_with\_associated\_documents.documents->  
 document=>  
 product\_quality\_report

### 5.1.15.17.8 inspection\_activity to inspection\_plan (as customer\_requirements)

AIM element: PATH  
Reference Path: inspection\_activity <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action.chosen\_method  
action=>  
product\_definition\_process=>  
inspection\_plan

### 5.1.15.18 Machining\_activity

AIM element: machining\_activity  
Source: ISO 10303-223  
Reference Path: machining\_activity  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

#### 5.1.15.18.1 Machining\_activity to external\_schema\_definition (as iso\_10303-240\_data)

AIM element: PATH  
Reference Path: machining\_activity  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=externally\_defined\_item  
externally\_defined\_item=>  
externally\_defined\_schema



### 5.1.15.19 Manufacturing\_process

AIM element: manufacturing\_process  
 Source: ISO 10303-223  
 Reference path: manufacturing\_process<=  
 action\_method

#### 5.1.15.19.1 description

AIM element: action\_method.description  
 Source: ISO 10303-41  
 Reference path: manufacturing\_process<=  
 action\_method  
 action\_method.description

#### 5.1.15.19.2 name

AIM element: action\_method.name  
 Source: ISO 10303-41  
 Reference path: manufacturing\_process<=  
 action\_method  
 action\_method.name

#### 5.1.15.19.3 manufacturing\_process to casting\_activity (as assigned - operation)

AIM element: PATH  
 Reference path: manufacturing\_process<=  
 action\_method  
 action\_method<-  
 action\_method\_relationship.relateing\_method  
 action\_method\_relationship  
 (action\_method\_relationship=>  
 serial\_action\_method=>  
 sequential\_method=>  
 manufacturing\_activity\_relationship)  
 action\_method\_relationship.related\_method->  
 action\_method=>  
 process\_plan\_activity=>  
 manufacturing\_activity=>  
 casting\_activity

#### 5.1.15.19.4 manufacturing\_process to setup\_activity (as setup)

AIM element: PATH  
Reference path: manufacturing\_process<=  
action\_method  
action\_method<-  
action\_method\_relationship.relating\_method  
action\_method\_relationship  
(action\_method\_relationship=>  
serial\_action\_method=>  
sequential\_method=>  
single\_activity\_relationship)  
action\_method\_relationship.related\_method->  
action\_method=>  
process\_plan\_activity=>  
(ancillary\_setup)  
(machine\_setup=>  
casting\_equipment\_setup)

#### 5.1.15.20 Metalcaster\_simulation

AIM element: simulation\_process  
Source: ISO 10303-223  
Reference Path: metalcaster\_simulation<=  
simulation\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

##### 5.1.15.20.1 metalcaster\_simulations to simulation\_exception\_report (as defect\_results)

AIM element: PATH  
Reference Path: metalcaster\_simulation<=  
simulation\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method=>

```

action_method_with_associated_documents
action_method_with_associated_documents.documents[i]->
document=>
document.kind->
document_type
document_type.product_data_type='simulation exception report'

```

### 5.1.15.20.2 metalcaster\_simulation to tooling\_report (as suggestion\_for\_-changes)

```

AIM element:    PATH
Reference Path: metalcaster_simulation<=
                simulation_process <=
                casting_activity<=
                manufacturing_activity<=
                process_plan_activity<=
                action_method=>
                action_method_with_associated_documents
                action_method_with_associated_documents.documents[i]->
                document=>
                document.kind->
                document_type
                document_type.product_data_type='simulation tooling report'

```

### 5.1.15.20.3 metalcaster\_simulation to gating\_system (as proposed\_rigging)

```

AIM element:    PATH
Reference Path: metalcaster_simulation<=
                simulation_process <=
                casting_activity<=
                manufacturing_activity<=
                process_plan_activity<=
                action_method<-
                action_method_assignment.assigned_action_method
                action_method_assignment=>
                applied_action_method_assignment
                applied_action_method_assignment.items[i]->
                action_method_item
                action_method_item=shape_aspect
                shape_aspect=>
                casting_instanced_feature<=
                casting_feature_definition=>
                ejector_design_system=>
                gating_system

```

#### **5.1.15.20.4 metalcaster\_simulation to pattern\_equipment\_available (as available\_tooling)**

AIM element: PATH  
Reference Path: metalcaster\_simulation<=  
simulation\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
customer\_casting\_requirement=>  
pattern\_equipment\_available

#### **5.1.15.21 Non\_casting\_operation**

AIM element: non\_casting\_operation  
Source: ISO 10303-223  
Reference path: non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

#### **5.1.15.21.1 non\_casting\_operation to descriptive\_parameter (as operation description)**

AIM element: PATH  
Reference path: non\_casting\_operation<=  
casting\_activity<=

```

manufacturing_activity<=
process_plan_activity<=
action_method
characterized_action_definition=action_method
characterized_action_definition<-
action_property.definition
action_property<-
action_property_representation.property
action_property_representation
action_property_representation.representation
representation
representation.items[i] ->
representation_item
representation_item.name='operation description'
descriptivre_representation_item

```

### 5.1.15.21.2 non\_casting\_operation to finishing\_or\_machining\_operations (as requirement)

```

AIM element:    PATH
Reference Path: heat_treat<=
                non_casting_operation<=
                casting_activity<=
                manufacturing_activity<=
                process_plan_activity<=
                action_method<-
                action_method_assignment.assigned_action_method
                action_method_assignment=>
                applied_action_method_assignment
                applied_action_method_assignment.items[i]->
                action_method_item
                action_method_item=property_definition
                property_definition=>
                finishing_or_machining_operations

```

### 5.1.15.21.3 non\_casting\_operation to cast\_part (as part\_geometry\_with\_machining\_allowance)

AIM element: PATH  
Reference Path: non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
product\_definition  
cast\_part

### 5.1.15.21.4 non\_casting\_operation to design\_part (as part\_geometry\_with\_no\_machining\_allowance)

AIM element: PATH  
Reference Path: non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
product\_definition  
design\_part

### 5.1.15.22 Non\_equipment\_process

AIM element: non\_casting\_equipment\_process  
Source: ISO 10303-240  
Reference path: non\_casting\_equipment\_process<=  
non\_machining\_process<=  
manufacturing\_process <=  
action\_method

### 5.1.15.23 Non\_permanent\_moulding\_process

AIM element: non\_permanent\_moulding\_process  
 action\_method  
 Source: ISO 10303-223  
 Reference Path: non\_permanent\_mould\_process <=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method

#### 5.1.15.23.1 moulding\_process

AIM element: PATH  
 Reference Path: non\_permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method  
 {(action\_method.description = 'shell')  
 (action\_method.description = 'green sand')  
 (action\_method.description = 'no bake')  
 (action\_method.description = 'lost form')  
 (action\_method.description = 'vacuum')}}

### 5.1.15.23.2 number\_of\_impressions

AIM element: PATH  
Reference Path: non\_permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='number of impressions'  
measure\_representation\_item <=  
measure\_with\_unit  
measure\_with\_unit.value\_component->  
measure\_value  
measure\_value=count\_measure  
count\_measure



### 5.1.15.23.3 pattern\_form

AIM element: PATH

Reference Path: non\_permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='pattern form'  
 descriptive\_representation\_item  
 {(descriptive\_representation\_item.description = 'mounted match plate')  
 (descriptive\_representation\_item.description = 'mounted cope and drag inserts')  
 (descriptive\_representation\_item.description = 'mounted cope and drag boards')  
 (descriptive\_representation\_item.description = 'mounted flaskless cope and drag')  
 (descriptive\_representation\_item.description = 'loose split pattern')  
 (descriptive\_representation\_item.description = 'loose pattern with follow block')  
 (descriptive\_representation\_item.description = 'sweep')};

#### 5.1.15.23.4 non\_permanent\_moulding\_process to flask (as flask\_type)

AIM element: PATH  
Reference Path: non\_permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
sand\_cast\_feature\_definition=>  
flask

#### 5.1.15.23.5 non\_permanent\_moulding\_process to mould\_box (as flask\_type)

AIM element: PATH  
Reference Path: non\_permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
sand\_cast\_feature\_definition=>  
mould\_box

**5.1.15.23.6 non\_permanent\_moulding\_process to flaskless (as flask\_type)**

AIM element: PATH  
 Reference Path: non\_permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item= property\_definition  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_object  
 characterized\_object=>  
 sand\_cast\_feature\_definition=>  
 flaskless

**5.1.15.23.7 non\_permanent\_moulding\_process to production\_pattern\_definition (as pattern)**

AIM element: PATH  
 Reference Path: non\_permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item= property\_definition  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_object  
 characterized\_object=>  
 sand\_cast\_feature\_definition=>  
 production\_pattern\_definition

### 5.1.15.24 Permanent\_moulding\_process

AIM element: permanent\_moulding\_process  
action\_method  
Source: ISO 10303-223  
Reference Path: permanent\_mould\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

#### 5.1.15.24.1 moulding\_process

AIM element: PATH  
Reference Path: permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method  
{(action\_method.consequence.description = 'static')  
(action\_method.consequence.description = 'tilt')  
(action\_method.consequence.description = 'low pressure')};

### 5.1.15.24.2 number\_of\_impressions

AIM element: PATH  
 Reference Path: permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method  
 characterized\_action\_definition=action\_method  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='number of impressions'  
 measure\_representation\_item <=  
 measure\_with\_unit  
 measure\_with\_unit.value\_component->  
 measure\_value  
 measure\_value=count\_measure  
 count\_measure

### 5.1.15.24.3 permanent\_mould\_process to die\_mould\_vent (as ejection\_definition)

AIM element: PATH  
Reference Path: permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
die\_feature\_definition=>  
die\_mould\_vent

### 5.1.15.24.4 permanent\_mould\_process to ejector\_system (as vent\_definition)

AIM element: PATH  
Reference Path: permanent\_moulding\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
ejector\_feature\_definition=>  
ejector\_system

### 5.1.15.24.5 permanent\_mould\_process to production\_die\_mould (as permanent\_die\_definition)

AIM element: PATH  
 Reference Path: permanent\_moulding\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item= property\_definition  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_object  
 characterized\_object=>  
 casting\_feature\_definition=>  
 production\_tool=>  
 die\_design\_feature=>  
 production\_die\_mould

### 5.1.15.24.6 permanent\_mould\_process to ingate (as metal\_flow\_system)

AIM element: PATH  
 Reference Path: permanent\_mould\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item= property\_definition  
 property\_definition  
 property\_definition.definition->  
 characterized\_definition  
 characterized\_definition=characterized\_object  
 characterized\_object=>  
 gating\_feature\_definition=>  
 ingate

### 5.1.15.24.7 permanent\_mould\_process to runner (as metal\_flow\_system)

AIM element: PATH  
Reference Path: permanent\_mould\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
gating\_feature\_definition=>  
runner

### 5.1.15.24.8 permanent\_mould\_process to sprue (as metal\_flow\_system)

AIM element: PATH  
Reference Path: permanent\_mould\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item= property\_definition  
property\_definition  
property\_definition.definition->  
characterized\_definition  
characterized\_definition=characterized\_object  
characterized\_object=>  
gating\_feature\_definition=>  
sprue



**5.1.15.25 Performance\_rate**

AIM element: (allowed\_time)  
 (production\_rate)  
 Source: ISO 10303-223  
 ISO 10303-223  
 Reference Path: (allowed\_time <=)  
 (production\_rate <=)  
 action\_property

**5.1.15.25.1 source**

AIM element: action\_property.description  
 Source: ISO 10303-41  
 Reference Path: (allowed\_time <=)  
 (production\_rate <=)  
 action\_property  
 action\_property.description

**5.1.15.26 Production\_rate**

AIM element: production\_rate  
 Source: ISO 10303-223  
 Reference Path: production\_rate <=  
 action\_property

**5.1.15.26.1 time\_per\_unit**

AIM element: time\_measure\_with\_unit  
 Source: ISO 10303-41  
 Reference Path: production\_rate <=  
 action\_property <=  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation ->  
 representation  
 representation.items [i] ->  
 {representation\_item.name='time per unit'}  
 representation\_item =>  
 measure\_representation\_item <=  
 measure\_with\_unit =>  
 time\_measure\_with\_unit

### 5.1.15.26.2 unit\_quantity

AIM element: measure\_with\_unit  
Source: ISO 10303-41  
Reference Path: production\_rate <=  
action\_property <=  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation ->  
representation  
representation.items [i] ->  
{representation\_item.name='unit\_quantity'}  
representation\_item =>  
measure\_representation\_item <=  
measure\_with\_unit

### 5.1.15.27 Production\_welding

AIM element: production\_welding  
Source: ISO 10303-223  
Reference Path: production\_welding<=  
non\_casting\_operation<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method

### 5.1.15.28 Setup\_activity

AIM element: (ancillary\_setup)  
(equipment\_setup)  
Source: ISO 10303-223  
ISO 10303-223  
Reference Path: (ancillary\_setup <=)  
(equipment\_setup<=  
machine\_setup<=)  
process\_plan\_activity<=  
action\_method

### 5.1.15.29 Simulation\_process

AIM element: simulation\_process  
 Source: ISO 10303-223  
 Reference Path: simulation\_process <=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method

#### 5.1.15.29.1 simulation\_process to design\_part (as part\_geometry)

AIM element: PATH  
 Reference Path: simulation\_process<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action\_method  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definition  
                   product\_definition  
                   design\_part

#### 5.1.15.29.2 simulation\_process to design\_part (as cast\_part)

AIM element: PATH  
 Reference Path: simulation\_process<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action\_method  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definition  
                   product\_definition  
                   cast\_part

### 5.1.15.29.3 simulation\_process to design\_part (as cast\_part\_with\_rigging)

AIM element: PATH  
Reference Path: simulation\_process<=  
casting\_activity<=  
process\_plan\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
product\_definition  
cast\_part\_with\_rigging

### 5.1.15.29.4 simulation\_process to customer\_casting\_requirement (as quality\_requirements)

AIM element: PATH  
Reference Path: simulation\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
customer\_casting\_requirement

### 5.1.15.29.5 simulation\_process to simulation\_run (as run\_simulation)

AIM element: PATH  
 Reference Path: simulation\_process<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method<-  
                   action.chosen\_method  
                   action=>  
                   executed\_action=>  
                   simulation\_run

### 5.1.15.29.6 simulation\_process to tooling(as tooling\_for\_simulation)

AIM element: PATH  
 Reference Path: simulation\_process<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action\_method  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=product\_definition  
                   product\_definition=>  
                   casting\_product\_definition

### 5.1.15.30 Surface\_finish

AIM element: surface\_finish  
 Source: ISO 10303-223  
 Reference Path: surface\_finish<=  
                   non\_casting\_operation<=  
                   casting\_activity<=  
                   manufacturing\_activity<=  
                   process\_plan\_activity<=  
  
                   action\_method

### 5.1.15.31 Tooling\_process

AIM element: tooling\_process  
Source: ISO 10303-223  
Reference Path: tooling\_process <=  
                  casting\_activity<=  
                  manufacturing\_activity<=  
                  process\_plan\_activity<=  
                  action\_method

#### 5.1.15.31.1 number\_of\_impressions

AIM element: measure\_representation\_item  
Source: ISO 10303-43  
Reference Path: tooling\_process <=  
                  casting\_activity<=  
                  manufacturing\_activity<=  
                  process\_plan\_activity<=  
                  action\_method  
                  characterized\_action\_definition=action\_method  
                  characterized\_action\_definition<-  
                  action\_property.definition  
                  action\_property<-  
                  action\_property\_representation.property  
                  action\_property\_representation  
                  action\_property\_representation.representation  
                  representation  
                  representation.items[i] ->  
                  representation\_item  
                  representation\_item.name='number of impressions'  
                  measure\_representation\_item <=  
                  measure\_with\_unit  
                  measure\_with\_unit.value\_component->  
                  measure\_value  
                  measure\_value=count\_measure  
                  count\_measure

**5.1.15.31.2 shrinkage\_factor**

AIM element:       measure\_representation\_item  
Source:             ISO 10303-43  
Reference Path:    tooling\_process <=  
                    casting\_activity<=  
                    manufacturing\_activity<=  
                    process\_plan\_activity<=  
                    action\_method  
                    characterized\_action\_definition=action\_method  
                    characterized\_action\_definition<-  
                    action\_property.definition  
                    action\_property<-  
                    action\_property\_representation.property  
                    action\_property\_representation  
                    action\_property\_representation.representation  
                    representation  
                    representation.items[i] ->  
                    representation\_item  
                    representation\_item.name='shrinkage factor'  
                    representation\_item =>  
                    measure\_representation\_item  
                    {measure\_representation\_item <=  
                    measure\_with\_unit =>  
                    length\_measure\_with\_unit}

### 5.1.15.31.3 process\_method

AIM element: PATH  
Reference Path: tooling\_process <=  
casting\_activity <=  
manufacturing\_activity <=  
process\_plan\_activity <=  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition <=  
action\_property.definition  
action\_property <=  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation  
representation  
representation.items[i] ->  
representation\_item  
{representation\_item.name='process method'}  
representation\_item=>  
descriptive\_representation\_item  
{(descriptive\_representation\_item.description = 'permanent mould')  
(descriptive\_representation\_item.description = 'non permanent mould')  
(descriptive\_representation\_item.description = 'general casting')}

### 5.1.15.31.4 tooling\_proces to core (as cores\_to\_make)

AIM element: PATH  
Reference Path: tooling\_process <=  
casting\_activity <=  
manufacturing\_activity <=  
process\_plan\_activity <=  
action\_method <=  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=shape\_aspect  
shape\_aspect=>  
instanced\_feature=>  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
core



**5.1.15.31.5 tooling\_proces to gating\_system (as required\_gating)**

AIM element: PATH  
 Reference Path: tooling\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=shape\_aspect  
 shape\_aspect=>  
 instanced\_feature=>  
 casting\_feature\_definition=>  
 gating\_design\_feature=>  
 gating\_system

**5.1.15.31.6 tooling\_process to customer\_casting\_requirement (as customer\_requirements)**

AIM element: PATH  
 Reference Path: tooling\_process<=  
 casting\_activity<=  
 manufacturing\_activity<=  
 process\_plan\_activity<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action\_method  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definition  
 property\_definition=>  
 customer\_casting\_requirement

### 5.1.15.31.7 tooling\_process to design\_part (as part\_geometry)

AIM element: PATH  
Reference Path: tooling\_process<=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_assignment.assigned\_action\_method  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
product\_definition  
design\_part

### 5.1.15.31.8 tooling\_process to tool\_verification (as quality\_assurance)

AIM element: PATH  
Reference Path: tooling\_process <=  
casting\_activity<=  
manufacturing\_activity<=  
process\_plan\_activity<=  
action\_method<-  
action\_method\_relationship.relating\_method  
action\_method\_relationship  
(action\_method\_relationship.name='quality assurance')  
action\_method\_relationship.related\_method->  
action\_method=>  
casting\_method=>  
tool\_verification

## 5.1.16 Process\_plan UoF

### 5.1.16.1 Alternate\_process\_plan

AIM element: process\_plan\_version  
Source: ISO 10303-223  
Reference path: process\_plan\_version<=  
product\_definition\_process<=  
action  
(id\_attribute\_select=action  
id\_attribute\_select<-  
id\_attribute.identified\_item  
id\_attribute  
id\_attribute.attribute\_value='alternate process plan')

### 5.1.16.1.1 alternate\_ranking

AIM element: action.name  
 Source: ISO 10303-41  
 Reference path: process\_plan\_version<=  
                   product\_definition\_process<=  
                   action  
                   action.name

### alternate\_process\_plan to process\_plan\_version (as plan\_substitute)

AIM element: PATH  
 Reference path: process\_plan\_version<=  
                   product\_definition\_process<=  
                   action<-  
                   action\_relationship.related\_action  
                   action\_relationship=>  
                   alternate\_plan\_relationship  
                   action\_relationship.relate\_action->  
                   action=>  
                   product\_definition\_process=>  
                   process\_plan\_version

### 5.1.16.2 Continuous\_process

AIM element: continuous\_process\_relationship  
 Source: ISO 10303-223  
 Reference path: continuous\_process\_relationship<=  
                   sequential\_method<=  
                   serial\_action\_method<=  
                   action\_method\_relationship

#### 5.1.16.2.1 process\_type

AIM element: action\_method\_relationship.description  
 Source: ISO 10303-49  
 Reference path: continuous\_process\_relationship<=  
                   sequential\_method<=  
                   serial\_action\_method<=  
                   action\_method\_relationship  
                   action\_method\_relationship.description  
                   {(action\_method\_relationship.description= 'serial')  
                   (action\_method\_relationship.description= 'batch')  
                   (action\_method\_relationship.description= 'serial and batch')} }

### 5.1.16.2.2 continuous\_process to manufacturing\_process (as related\_process)

AIM element: PATH  
Reference path: continuous\_process\_relationship<=  
sequential\_method<=  
serial\_action\_method<=  
action\_method\_relationship  
action\_method\_relationship.related\_method->  
action\_method=>  
manufacturing\_process

### 5.1.16.2.3 continuous\_process to manufacturing\_process (as relating\_process)

AIM element: PATH  
Reference path: continuous\_process\_relationship<=  
sequential\_method<=  
serial\_action\_method<=  
action\_method\_relationship  
action\_method\_relationship.relying\_method->  
action\_method=>  
manufacturing\_process

## 5.1.16.3 Process\_plan

AIM element: process\_plan  
Source: ISO 10303-223  
Reference Path: process\_plan\_version <=  
product\_definition\_process<=  
(product\_definition.identification='mini plan')  
action

### 5.1.16.3.1 description

AIM element: action.description  
Source: ISO 10303-41  
Reference path: process\_plan\_version<=  
product\_definition\_process<=  
action  
action.description

### 5.1.16.3.2 process\_plan\_identifier

AIM element:     product\_definition\_process.identification  
 Source:         ISO 10303-41  
 Reference Path:   process\_plan\_version<=  
                   product\_definition\_process<=  
                   product\_definition\_process.identification

### 5.1.16.3.3 process\_plan\_to\_activity\_execution\_result)

AIM element:     PATH  
 Reference path:   process\_plan\_version<=  
                   product\_definition\_process<=  
                   action<-  
                   action\_relationship.relatering\_action  
                   action\_relationship  
                   action\_relationship.related\_action->  
                   action=>  
                   executed\_action  
                   activity\_execution\_result

### 5.1.16.3.4 process\_plan\_to\_casting\_activity (as activities)

AIM element:     PATH  
 Reference path:   process\_plan\_version<=  
                   product\_definition\_process<=  
                   action  
                   action.chosen\_method->  
                   action\_method<-  
                   action\_method\_relationship.relatering\_method  
                   action\_method\_relationship  
                   action\_method\_relationship.related\_method->  
                   action\_method=>  
                   process\_plan\_activity=>  
                   manufacturing\_activity=>  
                   casting\_activity

### 5.1.16.3.5 process\_plan to part\_version(as defines\_manufacturing\_for)

AIM element: PATH  
Reference path: process\_plan\_version<=  
product\_definition\_process <-  
process\_product\_association.process  
process\_product\_association  
process\_product\_association.defined\_product->  
characterized\_product\_definition  
characterized\_product\_definition = product\_definition  
product\_definition

### 5.1.16.4 Process\_plan\_security

AIM element: process\_plan\_security  
Source: ISO 10303-223  
Reference path: process\_plan\_security=>  
security\_classification

#### 5.1.16.4.1 classification\_date

AIM element: calendar\_date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference path: process\_plan\_security  
date\_item = process\_plan\_security  
date\_item <-  
applied\_date\_assignment.items [i]  
applied\_date\_assignment <=  
date\_assignment  
{date\_assignment.role ->  
date\_role  
date\_role.name='classification date'}  
date\_assignment.assigned\_date ->  
date =>  
calendar\_date

### 5.1.16.4.2 declassification\_date

AIM element: calendar\_date  
 Source: ISO 10303-41  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Reference path: process\_plan\_security  
 date\_item = process\_plan\_security  
 date\_item <-  
 applied\_date\_assignment.items [i]  
 applied\_date\_assignment <=  
 date\_assignment  
 {date\_assignment.role ->  
 date\_role  
 date\_role\_name='declassification date'}  
 date\_assignment.assigned\_date ->  
 date =>  
 calendar\_date

### 5.1.16.4.3 security\_code

AIM element: security\_classification\_level.name  
 Source: ISO 10303-41  
 Rules: restrict\_security\_classification\_level – (See 5.2.4.28)  
 dependent\_instantiable\_security\_classification\_level – (See 5.2.4.13)  
 Reference path: process\_plan\_security=>  
 security\_classification  
 security\_classification.security\_level ->  
 security\_classification\_level  
 security\_classification\_level.name

### 5.1.16.4.4 process\_plan\_security to activity ( as identified\_by\_activity)

AIM element: PATH  
 Reference path: process\_plan\_security=>  
 security\_classification<-  
 security\_classification\_assignment.assigned\_security\_classification  
 security\_classification\_assignment=>  
 applied\_security\_classification\_assignment  
 applied\_security\_classification\_assignment.items [i]->  
 security\_classification\_item  
 security\_classification\_item=process\_plan\_activity  
 process\_plan\_activity

#### **5.1.16.4.5 process\_plan\_security to process\_plan\_version (as identified\_by\_-process\_plan)**

AIM element: PATH  
Reference path: process\_plan\_security=>  
security\_classification<-  
security\_classification\_assignment.assigned\_security\_classification  
security\_classification\_assignment=>  
applied\_security\_classification\_assignment  
applied\_security\_classification\_assignment.items [i]->  
security\_classification\_item  
security\_classification\_item =process\_plan\_version  
process\_plan\_version

#### **5.1.16.5 Process\_plan\_version**

AIM element: process\_plan\_version  
Source: ISO 10303-223  
Reference Path: process\_plan\_version <=  
product\_definition\_process<=  
action



### 5.1.16.5.1 process\_plan\_version to manufacturing\_process (as activities\_to\_produce\_part)

AIM element: PATH

Reference Path: process\_plan\_version <=  
 product\_definition\_process<=  
 action  
 action.chosen\_method->  
 action\_method  
 action\_method=>  
 manufacturing\_process  
 {manufacturing\_process=>  
 (non\_machining\_process)  
 (machining\_process)}  
 {manufacturing\_process<=  
 action\_method  
 action\_method<-  
 action\_method\_relationship.relatng\_method  
 action\_method\_relationship  
 (action\_method\_relationship=>  
 serial\_action\_method=>  
 sequential\_method=>  
 manufacturing\_process\_relationship)  
 action\_method\_relationship.related\_method->  
 action\_method=>  
 manufacturing\_process}

### 5.1.16.5.2 process\_plan\_version to material\_usage as amount of raw materials)

AIM element: PATH

Reference path: process\_plan\_version<=  
 product\_definition\_process<=  
 action<-  
 action\_assignment.assigned\_action  
 action\_assignment=>  
 applied\_action\_assignment  
 applied\_action\_assignment.items[i]->  
 action\_item  
 action\_item=property\_definition  
 property\_definition=>  
 material\_property=>  
 material\_usage

### 5.1.16.5.3 process\_plan\_version to ordered\_part (quantity\_of\_parts)

AIM element: PATH  
Reference path: process\_plan\_version<=  
product\_definition\_process<=  
action<-  
action\_assignment.assigned\_action  
action\_assignment=>  
ordered\_part

### 5.1.16.5.4 process\_plan\_version to property\_parameter (as process\_plan\_properties)

AIM element: PATH  
Reference Path: process\_plan\_version <=  
product\_definition\_process<=  
action  
characterized\_action\_definition=action  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation->  
representation  
representation.item[i]->  
representation\_item  
{representation\_item.name='process plan properties'}  
(descriptive\_representation\_item)  
(measure\_representation\_item)

### 5.1.16.5.5 process\_plan\_version to special\_instruction (as process\_plan\_information)

AIM element: PATH  
 Reference Path: process\_plan\_version <=  
 product\_definition\_process<=  
 action  
 characterized\_action\_definition=action  
 characterized\_action\_definition<-  
 action\_property.definition  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation->  
 representation  
 representation.items[i]->  
 representation\_item=>  
 {representation\_item.name='special instruction'}  
 descriptive\_representation\_item

### 5.1.16.6 Revision

AIM element: revision  
 Source: ISO 10303-223  
 Reference path: revision<=  
 action\_relationship

#### 5.1.16.6.1 reason\_for\_revision

AIM element: relationship\_condition.condition\_description  
 Source: ISO 10303-49  
 Reference path: revision<=  
 action\_relationship  
 relationship\_with\_condition=action\_relationship  
 relationship\_with\_condition<-  
 relationship\_condition.applicable\_relationships[i]  
 relationship\_condition  
 {relationship\_condition.name='reason for revision'}  
 relationship\_condition.condition\_description

### 5.1.16.6.2 description

AIM element: action\_relationship.description  
Source: ISO 10303-41  
Reference path: revision<=  
action\_relationship  
action\_relationship.description

### 5.1.16.6.3 revision\_level

AIM element: versioned\_action\_request.version  
Source: ISO 10303-41  
Reference path: revision<=  
action\_relationship  
action\_relationship.related\_action->  
action  
action.chosen\_method->  
action\_method<-  
action\_request\_solution.method  
action\_request\_solution  
{action\_request\_solution.description='revision'}  
action\_request\_solution.request->  
versioned\_action\_request  
versioned\_action\_request.version

### 5.1.16.6.4 revision to status\_authority ( as approved\_by)

AIM element: PATH  
Reference path: revision  
approval\_item = revision  
approval\_item<-  
applied\_approval\_assignment.items [i]  
applied\_approval\_assignment<=  
approval\_assignment  
approval\_assignment.assigned\_approval->  
approval

### 5.1.16.6.5 revision to process\_plan\_version (as related\_to)

AIM element: PATH  
Reference path: revision<=  
action\_relationship  
action\_relationship.related\_action->  
action =>  
product\_definition\_process =>  
process\_plan\_version

### 5.1.16.6.6 revision to process\_plan\_version (as relating\_to)

AIM element: PATH  
 Reference path: revision<=  
                   action\_relationship  
                   action\_relationship.relating\_action->  
                   action =>  
                   product\_definition\_process =>  
                   process\_plan\_version

## 5.1.17 Quality control UoF

### 5.1.17.1 Activity\_execution\_result

AIM element: activity\_execution\_result  
 Source: ISO 10303-223  
 Reference Path: activity\_execution\_result<=  
                   executed\_action

#### 5.1.17.1.1 description

AIM element: action.description  
 Source: ISO 10303-41  
 Reference Path: activity\_execution<=  
                   executed\_action<=  
                   action  
                   action.description

#### 5.1.17.1.2 activity\_execution\_result to activity\_result\_record (as activity\_results)

AIM element: PATH  
 Reference Path: activity\_execution\_result<=  
                   executed\_action<=  
                   action  
                   action\_action=characterized\_action\_definition  
                   characterized\_action\_definition <-  
                   action\_property.definition  
                   action\_property <-  
                   action\_property\_representation.property  
                   action\_property\_representation  
                   action\_property\_representation.representation ->  
                   representation=>  
                   activity\_result\_record

### 5.1.17.1.3 activity\_execution\_result to sampled\_set (as part\_measured)

AIM element: PATH  
Reference Path: activity\_execution\_result<=  
executed\_action<=  
action =>  
property\_process <-  
process\_property\_association.process  
process\_property\_association  
{process\_property\_association.description='validation part shape'}  
process\_property\_association.property\_or\_shape ->  
property\_or\_shape\_select=property\_definition  
property\_definition=>  
sampled\_set}

### 5.1.17.2 Activity\_result\_record

AIM element: activity\_result\_record  
Source: ISO 10303-223  
Reference Path: activity\_result\_record<=  
representation

#### 5.1.17.2.1 date\_and\_time

AIM element: calendar\_date  
Source: ISO 10303-41  
Reference Path: activity\_result\_record  
date\_and\_time\_item = activity\_result\_record  
date\_and\_time\_item <-  
applied\_dateand\_time\_assignment.items [i]  
applied\_date\_and\_time\_assignment <=  
date\_and\_time\_assignment  
{date\_and\_time\_assignment.role ->  
date\_time\_role  
date\_time\_role.name='production date'}  
date\_time\_assignment.assigned\_date\_and\_time ->  
date\_and\_time

### 5.1.17.3 Casting\_per\_order

AIM element: casting\_per\_order  
Source: ISO 10303-223  
Reference Path: versioned\_action\_request=>  
casting\_verification=>  
casting\_per\_order

### 5.1.17.3.1 casting\_per\_order to sampled\_set(as sampling\_of\_parts)

AIM element: PATH  
 Reference Path: casting\_per\_order<=  
 casting\_verification<=  
 versioned\_action\_request<-  
 action\_directive.request[i]  
 action\_directive  
 directed\_action.directive  
 directed\_action<=  
 executed\_action<=  
 action=>  
 property\_process<-  
 process\_property\_association.process  
 process\_property\_association  
 process\_property\_association.property\_or\_shape  
 property\_or\_shape=property\_definition  
 property\_definition=>  
 sampled\_set

### 5.1.17.4 Casting\_verification

AIM element: casting\_verification  
 Source: ISO 10303-223  
 Reference Path: versioned\_action\_request=>  
 casting\_verification

#### 5.1.17.4.1 **cavity\_impression\_id**

AIM element: descriptive\_representation\_item  
Source: ISO 10303-41  
Reference Path: casting\_verification<=  
versioned\_action\_request  
version\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='cavity impression id'  
descriptive\_representation\_item

#### 5.1.17.4.2 **casting\_verification\_to\_tolerance\_requirement(as verify\_with)**

AIM element: PATH  
Reference Path: casting\_verification  
versioned\_action\_request<-  
action\_request\_assignment.assigned\_action\_request  
action\_request\_assignment=>  
applied\_action\_request\_assignment  
applied\_action\_request\_assignment.items[i]->  
action\_request\_item  
action\_request\_item= property\_definition  
property\_definition=>  
tolerance\_requirement



### 5.1.17.4.3 casting\_verification to quality\_acceptance\_report (as destruction)

AIM element: PATH  
 Reference Path: casting\_verification  
 versioned\_action\_request<-  
 action\_request\_assignment.assigned\_action\_request  
 action\_request\_assignment=>  
 applied\_action\_request\_assignment  
 applied\_action\_request\_assignment.items[i]->  
 action\_request\_item  
 action\_request\_item=document  
 document  
 (document.description='destruction')  
 document.kind->  
 document\_type  
 document\_type.product\_data\_type='quality\_acceptance\_report'

### 5.1.17.4.4 casting\_verification to quality\_acceptance\_report (as nde)

AIM element: PATH  
 Reference Path: casting\_verification  
 versioned\_action\_request<-  
 action\_request\_assignment.assigned\_action\_request  
 action\_request\_assignment=>  
 applied\_action\_request\_assignment  
 applied\_action\_request\_assignment.items[i]->  
 action\_request\_item  
 action\_request\_item=document  
 document  
 (document.description='nde')  
 document.kind->  
 document\_type  
 document\_type.product\_data\_type='quality\_acceptance\_report'

### 5.1.17.5 Checking\_aid\_tool

AIM element: checking\_aid\_tool  
 Source: ISO 10303-223  
 Reference Path: checking\_aid\_tool<=  
 customer\_casting\_requirement  
 property\_definition

### 5.1.17.5.1 aid\_id

AIM element: PATH  
Reference Path: checking\_aid\_tool<=  
customer\_casting\_requirement  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='aid id'  
descriptive\_representation\_item

### 5.1.17.5.2 aid\_name

AIM element: PATH  
Reference Path: checking\_aid\_tool<=  
customer\_casting\_requirement  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='aid name'  
descriptive\_representation\_item

**5.1.17.5.3 aid\_tool**

AIM element: PATH  
 Reference Path: checking\_aid\_tool<=  
 customer\_casting\_requirement  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<=  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='aid tool'  
 descriptive\_representation\_item  
 (descriptive\_representation\_item.name='fixture')  
 (descriptive\_representation\_item.name='guage')  
 (descriptive\_representation\_item.name='paper document')  
 (descriptive\_representation\_item.name='template')

**5.1.17.5.4 revision\_id**

AIM element: PATH  
 Reference Path: checking\_aid\_tool<=  
 customer\_casting\_requirement  
 property\_definition  
 represented\_definition=property\_definition  
 represented\_definition<=  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name='revision id'  
 descriptive\_representation\_item

### 5.1.17.5.5 checking\_aid\_tool to date ( as revision\_date)

AIM element: PATH  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: checking\_aid\_tool<=  
property\_definition  
date\_item=property\_definition  
date\_item<-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment  
applied\_date\_assignment<=  
date\_assignment  
date\_assignment.assigned\_date->  
date

### 5.1.17.6 Core\_assembly

AIM element: core\_assembly  
Source: ISO 10303-223  
Reference Path: action\_method=>  
casting\_method=>  
core\_assembly

#### 5.1.17.6.1 core\_assembly to descriptive\_parameter (as hard\_form)

AIM element: PATH  
Reference Path: core\_assembly<=  
casting\_method<=  
action\_method<-  
characterized\_action\_definition=action\_method  
characterized\_action\_definition<-  
action\_property.definition  
action\_property<-  
action\_property\_representation.property  
action\_property\_representation  
action\_property\_representation.representation->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='hard form'  
descriptive\_representation\_item

### 5.1.17.6.2 core\_assembly to proof\_report (as soft\_copy)

AIM element: PATH  
 Reference Path: core\_assembly<=  
                   casting\_method<=  
                   action\_method=>  
                   action\_method\_with\_associated\_document  
                   action\_method\_with\_associated\_document.documents[i]->  
                   document  
                   document.kind->  
                   document\_type  
                   document\_type.product\_data\_type='proof report'

### 5.1.17.7 Composition\_element\_record

AIM element: composition\_element\_record  
 Source: ISO 10303-223  
 Reference Path: composition\_element\_record =>  
                   activity\_result\_record<=  
                   representation

#### 5.1.17.7.1 composition\_element\_record to numeric\_parameter (as actual\_composition\_amount)

AIM element: PATH  
 Reference Path: composition\_element\_record =>  
                   activity\_result\_record<=  
                   representation  
                   representation.items[i] ->  
                   {representation\_item  
                   representation\_item.name = 'actual\_composition amount'}  
                   representation\_item =>  
                   measure\_representation\_item  
                   {measure\_representation\_item <=  
                   measure\_with\_unit =>  
                   length\_measure\_with\_unit}

### **5.1.17.7.2 composition\_element\_record to composition\_element (as recorded\_usage)**

AIM element: PATH  
Reference Path: composition\_element\_record =>  
activity\_result\_record<=  
representation<-  
property\_definition\_representation.used\_representation  
property\_definition\_representation  
property\_definition\_representation.definition->  
property\_definition=>  
material\_property=>  
composition\_element

### **5.1.17.8 Composition\_inspection**

AIM element: composition\_inspection  
Source: ISO 10303-41  
Reference Path: composition\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition

#### **5.1.17.8.1 composition\_inspection to material (as composition\_result)**

AIM element: product\_definition  
Source: ISO 10303-41  
Reference Path: composition\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition  
property\_definition.definition ->  
characterized\_definition  
characterized\_definition=characterized\_product\_definition  
characterized\_product\_definition=product\_definition  
product\_definition

#### **5.1.17.8.2 composition\_inspection to material\_property (as requirements)**

AIM element: PATH  
Reference Path: composition\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition<-

```

property_definition_relationship.relatering_property_definition
property_definition_relationship
property_definition_relationship.related_property_definition->
property_definition=>
material_property

```

### 5.1.17.9 Equipment\_setting\_record

AIM element: equipment\_setting\_record  
Source: ISO 10303-223  
Reference Path: equipment\_setting\_record =>  
activity\_result\_record<=  
representation

#### 5.1.17.9.1 equipment\_setting\_record to property\_parameter (as actual\_-machine\_setting)

AIM element: PATH  
Reference Path: equipment\_setting\_record =>  
activity\_result\_record<=  
representation  
representation.items [i]->  
representation\_item =>  
representation\_item.name= 'actual parameter'  
(measure\_representation\_item)  
(descriptive\_representation\_item)

#### 5.1.17.9.2 equipment\_setting\_record to equipment\_usage (as recorded\_-usage)

AIM element: PATH  
Reference Path: equipment\_setting\_record =>  
activity\_result\_record<=  
representation<-  
resource\_property\_representation.representation  
resource\_property\_representation  
resource\_property\_representation.property->  
resource\_property  
resource\_property.resource->  
characterized\_resource\_definition  
characterized\_resource\_definition= action\_resource\_requirement  
action\_resource\_requirement=>  
requirement\_for\_action\_resource=>  
equipment\_usage

### 5.1.17.10 Equipment\_usage

AIM element: equipment\_usage  
Source: ISO 10303-223  
R Reference Path: action\_resource\_requirement=>  
                  requirement\_for\_action\_resource=>  
                  equipment\_usage

#### 5.1.17.10.1 equipment\_usage to equipment (as equipment\_usage\_context)

AIM element: PATH  
Reference path: equipment\_usage<=  
                  requirement\_for\_action\_resource<=  
                  requirement\_for\_action\_resource.resources[i]->  
                  action\_resource =>  
                  machine=>  
                  casting\_equipment

#### 5.1.17.10.2 equipment\_usage to property\_parameter (as equipment\_parameter)

AIM element: PATH  
Reference Path: equipment\_usage<=  
                  requirement\_for\_action\_resource<=  
                  action\_resource\_requirement  
                  characterized\_resource\_definition= action\_resource\_requirement  
                  characterized\_resource\_definition<-  
                  resource\_property.resource  
                  resource\_property<-  
                  resource\_property\_representation.property  
                  resource\_property\_representation  
                  resource\_property\_representation.representation  
                  representation  
                  representation.items[i] ->  
                  representation\_item  
                  (descriptive\_representation\_item)  
                  (measure\_representation\_item)

### 5.1.17.11 First\_article

AIM element: first\_article  
Source: ISO 10303-223  
Reference Path: versioned\_action\_request=>  
                  casting\_verification=>  
                  first\_article



**5.1.17.11.1 number\_of\_pieces**

AIM element: PATH  
 Reference Path: first\_article  
 casting\_verification  
 versioned\_action\_request<-  
 action\_directive.request[i]  
 action\_directive  
 directed\_action.directive  
 directed\_action<=  
 executed\_action<=  
 action  
 characterized\_action\_definition=action<-  
 action\_property.definition  
 action\_property  
 action\_property<-  
 action\_property\_representation.property  
 action\_property\_representation  
 action\_property\_representation.representation->  
 representation  
 representation.items[i] ->  
 representation\_item  
 representation\_item.name = 'number of pieces'  
 representation\_item =>  
 measure\_representation\_item  
 measure\_with\_unit  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

**5.1.17.11.2 first\_article to approval (as customer\_approval)**

AIM element: PATH  
 Reference Path: first\_article  
 approval\_item= first\_article  
 approval\_item<-  
 applied\_approval\_assignment.items[i]  
 applied\_approval\_assignment<=  
 approval\_assignment  
 {approval\_assignment.role->  
 object\_role  
 object\_role.name='customer approval'}  
 approval\_assignment.assigned\_approval->  
 approval

### 5.1.17.11.3 first\_article to date (as delivery\_date)

AIM element: PATH  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: first\_article  
date\_item=first\_article  
date\_item<-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment<=  
date\_assignment  
{date\_assignment.role->  
date\_role  
date\_role.name='delivery date'}  
date\_assignment.assigned\_date->  
date

### 5.1.17.11.4 first\_article to master\_sample (as customer\_approved)

AIM element: PATH  
Reference Path: first\_article  
casting\_verification  
versioned\_action\_request<-  
action\_directive.request[i]  
action\_directive  
directed\_action.directive  
directed\_action<=  
executed\_action<=  
action<-  
action\_assignment.assigned\_action  
action\_assignment=>  
applied\_action\_assignment  
applied\_action\_assignment.items[i]->  
action\_item  
action\_item=product\_definition  
cast\_part  
master\_sample

**5.1.17.12 Heat**

AIM element: dated\_effectivity  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity<=  
 effectivity  
 name\_attribute\_select=effectivity  
 name\_attribute\_select<=  
 name\_attribute.named\_item  
 name\_attribute  
 name\_attribute.attribute\_value='heat'

**5.1.17.12.1 heat\_number**

AIM element: dated\_effectivity  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity<=  
 effectivity  
 effectivity.id

**5.1.17.12.2 heat to date(as production\_date)**

AIM element: date  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity  
 dated\_effectivity.effectivity\_start\_date->  
 date\_time\_or\_event\_occurrence  
 date\_time\_or\_event\_occurrence=date\_time\_select  
 date\_time\_select  
 date\_time\_select=date  
 date

**5.1.17.13 Inspection\_or\_test\_result**

AIM element: inspection\_or\_test\_result  
 Source: ISO 10303-41  
 Reference Path: property\_definition=>  
 inspection\_or\_test\_result  
 (composition\_inspection)  
 (tolerance\_inspection)  
 (other\_inspection)  
 (property\_inspection)

### 5.1.17.13.1 description

AIM element: property\_definition.description  
Source: ISO 10303-41  
Reference Path: inspection\_or\_test\_result<=  
property\_definition  
property\_definition.description

### 5.1.17.13.2 result

AIM element: descriptive\_representation\_item  
Source: ISO 10303-41  
Reference Path: inspection\_or\_test\_result<=  
property\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
representation\_item.name='result'  
(descriptive\_representation\_item)

### 5.1.17.13.3 inspection\_or\_test\_result to checking\_aid\_tool (as aid\_used)

AIM element: PATH  
Reference Path: inspection\_or\_test\_result<=  
property\_definition<-  
property\_definition\_relationship.relatering\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
checking\_aid\_tool

### 5.1.17.13.4 inspection\_or\_test\_result to shape\_aspect(as applies\_to)

AIM element: PATH  
Reference Path: inspection\_or\_test\_result<=  
property\_definition<-  
property\_definition\_relationship.relatering\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect

**5.1.17.13.5 inspection\_or\_test\_result to sampled\_set(as belongs\_to)**

AIM element: PATH  
 Reference Path: inspection\_or\_test\_result<=  
 property\_definition<-  
 property\_definition\_relationship.relatng\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 sampled\_set

**5.1.17.14 Lot**

AIM element: dated\_effectivity  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity<=  
 effectivity  
 name\_attribute\_select=effectivity  
 name\_attribute\_select<-  
 name\_attribute.named\_item  
 name\_attribute  
 name\_attribute.attribute\_value='lot'

**5.1.17.14.1 lot\_number**

AIM element: dated\_effectivity  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity<=  
 effectivity  
 effectivity.id

**5.1.17.14.2 lot to date(as production\_date)**

AIM element: date  
 Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
 Source: ISO 10303-41  
 Reference Path: dated\_effectivity  
 dated\_effectivity.effectivity\_start\_date->  
 date\_time\_or\_event\_occurrence  
 date\_time\_or\_event\_occurrence=date\_time\_select  
 date\_time\_select  
 date\_time\_select=date  
 date

### 5.1.17.15 **Material\_usage**

AIM element: material\_usage  
Source: ISO 10303-223  
R Reference Path: property\_definition=>  
material\_usage

#### 5.1.17.15.1 **Material\_usage to numeric\_parameter (as amount\_used)**

AIM element: PATH  
Reference Path: material\_usage<=  
property\_definition  
property\_definition = represented\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
{property\_definition\_representation=>  
material\_property\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i]->  
representation\_item  
(representation\_item.name='amount used')  
measure\_representation\_item

#### 5.1.17.15.2 **material\_usage to material (as material\_used)**

AIM element: PATH  
Reference Path: material\_usage<=  
property\_definition  
property\_definition.definition<-  
characterized\_definition  
characterized\_definition = characterized\_product\_definition  
characterized\_product\_definition  
characterized\_product\_definition = product\_definition  
product\_definition

### 5.1.17.16 **Material\_usage\_record**

AIM element: material\_usage\_record  
Source: ISO 10303-223  
Reference Path: material\_usage\_record=>  
activity\_result\_record<=  
representation

### 5.1.17.16.1 material\_usage\_record to numeric\_parameter (as actual\_substance\_usage\_amount)

AIM element: PATH  
 Reference Path: material\_usage\_record=>  
 activity\_result\_record<=  
 representation  
 representation.items [i]->  
 representation\_item =>  
 (representation\_item.name='actual substance usage amount')  
 measure\_representation\_item

### 5.1.17.16.2 material\_usage\_record to material\_usage (as recorded\_usage)

AIM element: PATH  
 Reference Path: material\_usage\_record =>  
 activity\_result\_record<=  
 representation<-  
 property\_definition\_representation.used\_representation  
 property\_definition\_representation  
 property\_definition\_representation.definition->  
 property\_definition=>  
 material\_usage

### 5.1.17.17 Process\_parameter\_record

AIM element: process\_parameter\_record  
 Source: ISO 10303-223  
 Reference Path: process\_parameter\_record<=  
 activity\_result\_record<=  
 representation

#### 5.1.17.17.1 process\_parameter\_record to property\_parameter (as actual\_parameter)

AIM element: PATH  
 Reference Path: process\_parameter\_record<=  
 activity\_result\_record<=  
 representation  
 representation.items [i]->  
 representation\_item =>  
 representation\_item.name= 'actual parameter'  
 (measure\_representation\_item)\  
 (descriptive\_representation\_item)

### 5.1.17.17.2 process\_parameter\_record to numeric\_parameter (as planned\_parameter)

AIM element: PATH  
Reference Path: process\_parameter\_record<=  
activity\_result\_record<=  
representation  
representation.items [i]->  
representation\_item =>  
representation\_item.name= 'planned parameter'  
(measure\_representation\_item)  
(descriptive\_representation\_item)

### 5.1.17.18 Property\_inspection

AIM element: property\_inspection  
Source: ISO 10303-213  
Reference Path: property\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition

#### 5.1.17.18.1 Property\_inspection to property\_parameter (as property\_result)

AIM element: PATH  
Reference Path: property\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition <=  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
(representation\_item.name= 'property result')  
(descriptive\_representation\_item)  
(measure\_representation\_item)



### 5.1.17.18.2 property\_inspection to property\_parameter (as requirements)

AIM element: PATH  
 Source: ISO 10303-41  
 Reference Path: property\_inspection<=  
 inspection\_or\_test\_result<=  
 property\_definition <-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i] ->  
 representation\_item  
 (representation\_item.name= 'requirements')  
 (descriptive\_representation\_item)  
 (measure\_representation\_item)

### 5.1.17.19 Sampled\_set

AIM element: sampled\_set  
 Source: ISO 10303-223  
 Reference Path: property\_definition=>  
 sampled\_set

#### 5.1.17.19.1 size

AIM element: lot\_effectivity.effectivity\_lot\_size  
 Source: ISO 10303-41  
 Reference Path: property\_definition  
 property\_definition <-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i] ->  
 representation\_item  
 (representation\_item.name= 'size')  
 measure\_representation\_item

### 5.1.17.19.2 sampled\_set to master\_sample(as master\_sample\_inspected)

AIM element: PATH  
Reference Path: sampled\_set<=  
property\_definition  
property\_definition.definition ->  
characterized\_definition  
characterized\_definition=characterized\_product\_definition  
characterized\_product\_definition=product\_definition  
product\_definition=>  
master\_sample

### 5.1.17.19.3 sampled\_set to heat(as sampled\_form)

AIM element: PATH  
Reference Path: sampled\_set<=  
property\_definition  
property\_definition=effectivity\_item  
effectivity\_item<-  
applied\_effectivity\_assignment.items[i]  
applied\_effectivity\_assignment<=  
effectivity\_assignment  
effectivity\_assignment.assigned\_effectivity->  
effectivity=>  
dated\_effectivity

### 5.1.17.19.4 sampled\_set to lot(as sampled\_form)

AIM element: PATH  
Reference Path: sampled\_set<=  
property\_definition  
property\_definition=effectivity\_item  
effectivity\_item<-  
applied\_effectivity\_assignment.items[i]  
applied\_effectivity\_assignment<=  
effectivity\_assignment  
effectivity\_assignment.assigned\_effectivity->  
effectivity=>  
lot\_effectivity

### 5.1.17.19.5 **sampled\_set to product\_quality\_report (as quality\_assurance\_ - document)**

AIM element: PATH  
 Reference Path: sampled\_set<=  
                   property\_definition  
                   document\_reference\_item=property\_definition  
                   document\_reference\_item<-  
                   applied\_document\_reference.items [i]  
                   applied\_document\_reference<=  
                   document\_reference  
                   document\_reference.assigned\_document->  
                   document  
                   (document.kind->  
                   document\_type  
                   document\_type.product\_data\_type='quality assurance document')

### 5.1.17.20 **Shape\_dimension**

AIM element: shape\_dimension  
 Source: ISO 10303-223  
 Reference Path: property\_definition=>  
                   shape\_dimension

#### 5.1.17.20.1 **shape\_dimension to dimensional\_tolerance(as dimensionals)**

AIM element: PATH  
 Reference Path: shape\_dimension<=  
                   property\_definition  
                   represented\_definition=property\_definition  
                   represented\_definition<-  
                   property\_definition\_representation.definition  
                   property\_definition\_representation  
                   {property\_definition\_representation=shape\_definition\_representation}  
                   property\_definition\_representation.used\_representation->  
                   representation  
                   shape\_representation=>  
                   shape\_dimension\_representation

### 5.1.17.20.2 shape\_dimension to production\_die\_mould (as tooling)

AIM element: PATH  
Reference Path: shape\_dimension <=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect=>  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
production\_tool=>  
die\_design\_feature=>  
production\_die\_mould

### 5.1.17.20.3 shape\_dimension to production\_core\_box (as tooling)

AIM element: PATH  
Reference Path: shape\_dimension <=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect=>  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
sand\_cast\_design\_feature=>  
production\_core\_box

**5.1.17.20.4 shape\_dimension to production\_pattern\_definition (as tooling)**

AIM element: PATH  
 Reference Path: shape\_dimension <=  
 property\_definition<-  
 property\_definition\_relationship.relating\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 product\_definition\_shape<-  
 shape\_aspect.of\_shape  
 shape\_aspect=>  
 casting\_instanced\_feature<=  
 casting\_feature\_definition=>  
 sand\_cast\_design\_feature=>  
 production\_pattern\_definition

**5.1.17.20.5 shape\_dimension to production\_die\_cast\_mould(as tooling)**

AIM element: PATH  
 Reference Path: shape\_dimension <=  
 property\_definition<-  
 property\_definition\_relationship.relating\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 product\_definition\_shape<-  
 shape\_aspect.of\_shape  
 shape\_aspect=>  
 casting\_instanced\_feature<=  
 casting\_feature\_definition=>  
 production\_tool=>  
 die\_design\_feature=>  
 production\_die\_cast\_mould

### 5.1.17.20.6 shape\_dimension to production\_investment\_cast\_mould (as tooling)

AIM element: PATH  
Reference Path: shape\_dimension <=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
product\_definition\_shape<-  
shape\_aspect.of\_shape  
shape\_aspect=>  
casting\_instanced\_feature<=  
casting\_feature\_definition=>  
production\_tool=>  
investment\_design\_feature=>  
production\_investment\_cast\_mould

### 5.1.17.21 Other\_inspection

AIM element: other\_inspection  
Source: ISO 10303-41  
Reference Path: other\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition

#### 5.1.17.21.1 inspection\_result

AIM element: PATH  
Reference Path: property\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition  
represented\_definition=property\_definition  
represented\_definition<-  
property\_definition\_representation.definition  
property\_definition\_representation  
property\_definition\_representation.used\_representation ->  
representation  
representation.items[i] ->  
representation\_item  
(representation\_item.name='inspection result')  
descriptive\_representation\_item

### 5.1.17.21.2 other\_inspection to inspection\_or\_test\_requirement (as requirements)

AIM element: property\_definition.description  
 Source: ISO 10303-41  
 Reference Path: other\_inspection<=  
 inspection\_or\_test\_result<=  
 property\_definition<-  
 property\_definition\_relationship.relatng\_property\_definition  
 property\_definition\_relationship  
 property\_definition\_relationship.related\_property\_definition->  
 property\_definition=>  
 inspection\_or\_test\_requirement

### 5.1.17.22 Tolerance\_inspection

AIM element: tolerance\_inspection  
 Source: ISO 10303-223  
 Reference Path: tolerance\_inspection<=  
 inspection\_or\_test\_result<=  
 property\_definition

#### 5.1.17.22.1 tolerance\_inspection to numeric\_parameter (as tolerance\_result)

AIM element: PATH  
 Reference Path: tolerance\_inspection<=  
 inspection\_or\_test\_result<=  
 property\_definition<-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i] ->  
 representation\_item  
 (representation\_item.name='tolerance result')  
 measure\_representation\_item

### **5.1.17.22.2 tolerance\_inspection to tolerance\_requirement (as requirements)**

AIM element: property\_definition.description  
Source: ISO 10303-41  
Reference Path: tolerance\_inspection<=  
inspection\_or\_test\_result<=  
property\_definition<-  
property\_definition\_relationship.relating\_property\_definition  
property\_definition\_relationship  
property\_definition\_relationship.related\_property\_definition->  
property\_definition=>  
tolerance\_requirement

### **5.1.17.23 Tool\_verification**

AIM element: tool\_verification  
Source: ISO 10303-223  
Reference Path: action\_method=>  
casting\_method=>  
tool\_verification

#### **5.1.17.23.1 tool\_verification to core\_assembly (as core\_fit)**

AIM element: PATH  
Reference Path: tool\_verification<=  
casting\_method<=  
action\_method<-  
action\_method\_relationship.relating\_method  
action\_method\_relationship  
action\_method\_relationship.related\_method  
action\_method=>  
casting\_method=>  
core\_assembly



### 5.1.17.23.2 tool\_verification to core\_print (as verify\_clearance)

AIM element: PATH  
 Reference Path: tool\_verification<=  
   casting\_method<=  
   action\_method<=  
   action\_method\_assignment.assigned\_action\_method  
   action\_method\_assignment =>  
   applied\_action\_method\_assignment  
   applied\_action\_method\_assignment.items[i] ->  
   action\_method\_item  
   action\_method\_item =shape\_aspect  
   shape\_aspect=>  
   casting\_instanced\_feature=>  
   casting\_feature\_definition=>  
   sand\_cast\_design\_feature=>  
   core\_print

### 5.1.17.23.3 tool\_verification to design\_part (as based\_on\_part\_design)

AIM element: PATH  
 Reference Path: tool\_verification<=  
   casting\_method<=  
   action\_method<=  
   action\_method\_assignment.assigned\_action\_method  
   action\_method\_assignment  
   action\_method\_assignment=>  
   applied\_applied\_action\_method\_assignment  
   applied\_applied\_action\_method\_assignment.items[i]->  
   action\_method\_item  
   action\_method\_item=product\_definition  
   product\_definition=>  
   design\_part

#### 5.1.17.23.4 tool\_verification to shape\_dimensions (as verify)

AIM element: PATH  
Reference Path: tool\_verification<=  
casting\_method<=  
action\_method<-  
applied\_action\_method\_assignment.assigned\_action\_method  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment=>  
applied\_applied\_action\_method\_assignment  
applied\_applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
shape\_dimensions

#### 5.1.17.23.5 tool\_verification to tolerance\_requirement (as customer\_requirement)

AIM element: PATH  
Reference Path: tool\_verification<=  
casting\_method<=  
action\_method<-  
applied\_action\_method\_assignment.assigned\_action\_method  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment=>  
applied\_applied\_action\_method\_assignment  
applied\_applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition=>  
tolerance\_requirement

### 5.1.18 Requisitions UoF

#### 5.1.18.1 Die\_mould\_requisition

AIM element: document  
Source: ISO 10303-41  
Reference Path: {document  
document.kind ->  
document\_type  
document\_type.product\_data\_type = 'die mould stock requisition'}

### 5.1.18.2 Indirect\_stock\_requisition

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: {document  
 document.kind ->  
 document\_type  
 document\_type.product\_data\_type = 'indirect stock requisition'}

### 5.1.18.3 Material\_requisition

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: {document  
 document.kind ->  
 document\_type  
 document\_type.product\_data\_type = 'material stock requisition'}

### 5.1.18.4 Pattern\_tooling\_requisition

AIM element: document  
 Source: ISO 10303-41  
 Reference Path: {document  
 document.kind ->  
 document\_type  
 document\_type.product\_data\_type = 'pattern tooling requisition'}

### 5.1.18.5 Requisition

AIM element: document  
 Source: ISO 10303-41

#### 5.1.18.5.1 quantity\_ordered

AIM element: document.name  
 Source: ISO 10303-41

#### 5.1.18.5.2 requisition\_description

AIM element: document.description  
 Source: ISO 10303-41

### 5.1.18.5.3 requisition\_number

AIM element: document.id  
Source: ISO 10303-41

### 5.1.18.5.4 requisition to date (as required\_delivery\_date)

AIM element: date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: document=>  
document\_with\_class  
date\_item = document\_with\_class  
date\_item <-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment <=  
{date\_assignment  
date\_assignment.role ->  
date\_role  
date\_role.name = 'required delivery date'}  
date\_assignment  
date\_assignment.assigned\_date ->  
date

### 5.1.18.5.5 requisition to date (as requisition\_date)

AIM element: date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: document=>  
document\_with\_class  
date\_item = document\_with\_class  
date\_item <-  
applied\_date\_assignment.items[i]  
applied\_date\_assignment <=  
{date\_assignment  
date\_assignment.role ->  
date\_role  
date\_role.name = 'requisition date'}  
date\_assignment  
date\_assignment.assigned\_date ->  
date

## 5.1.19 Shape\_representation\_for\_castings UoF

### 5.1.19.1 Base\_shape

AIM element: product\_definition\_shape  
 Source: ISO 10303-41  
 Reference path: product\_definition\_shape<=  
   {property\_definition  
   represented\_definition = property\_definition  
   represented\_definition <=  
   property\_definition\_representation.definition  
   property\_definition\_representation  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation.used\_representation ->  
   representation =>  
   (shape\_representation)  
   (shape\_representation\_with\_parameters)}

### 5.1.19.2 Casting\_envelope\_requirement

AIM element: casting\_envelope\_requirement  
 Source: ISO 10303-223  
 Reference Path: casting\_envelope\_requirement<=  
   product\_definition\_shape<=  
   property\_definition

### 5.1.19.2.1 casting\_envelope\_requirement to numeric\_parameter(as part\_height)

AIM element: PATH

Reference Path: casting\_envelope\_requirement<=  
product\_definition\_shape<=  
{property\_definition  
represented\_definition = property\_definition  
represented\_definition <=  
property\_definition\_representation.definition  
property\_definition\_representation  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
(representation =>  
shape\_representation)  
(shape\_representation\_with\_parameters)}  
representation.items[i] ->  
representation\_item  
representation\_item.name=' part height'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.19.2.2 casting\_envelope\_requirement to numeric\_parameter(as part\_length)

AIM element: PATH

Reference Path: casting\_envelope\_requirement<=  
product\_definition\_shape<=  
{property\_definition  
represented\_definition = property\_definition  
represented\_definition <-  
property\_definition\_representation.definition  
property\_definition\_representation  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
(representation =>  
shape\_representation)  
(shape\_representation\_with\_parameters)}  
representation.items[i] ->  
representation\_item  
representation\_item.name=' part length'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.19.2.3 casting\_envelope\_requirement to numeric\_parameter(as part\_width)

AIM element: PATH  
Reference Path: casting\_envelope\_requirement<=  
product\_definition\_shape<=  
{property\_definition  
represented\_definition = property\_definition  
represented\_definition <-  
property\_definition\_representation.definition  
property\_definition\_representation  
{property\_definition\_representation =>  
shape\_definition\_representation}  
property\_definition\_representation.used\_representation ->  
representation  
(representation =>  
shape\_representation)  
(shape\_representation\_with\_parameters)}  
representation.items[i] ->  
representation\_item  
representation\_item.name=' part width'  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.19.3 Curve\_point

AIM element: cartesian\_point  
Source: ISO 10303-42

#### 5.1.19.3.1 x\_value

AIM element: cartesian\_point.coordinates[1:1]  
Source: ISO 10303-42

#### 5.1.19.3.2 y\_value

AIM element: cartesian\_point.coordinates[2:2]  
Source: ISO 10303-42



### 5.1.19.4 Direction\_element

AIM element: direction\_shape\_representation  
 Source: ISO 10303-522  
 Reference Path: direction\_shape\_representation <=  
 shape\_representation <=  
 representation  
 {representation.items[i]->  
 representation\_item=>  
 geometric\_representation\_item=>  
 direction}

### 5.1.19.5 Drafted\_surface

AIM element: drafted\_surface  
 Source: ISO 10303-223  
 Reference Path: drafted\_surface <=  
 shape\_representation <=  
 representation

#### 5.1.19.5.1 drafted\_surface to numeric\_parameter(as draft\_amount)

AIM element: PATH  
 Reference Path: drafted\_surface <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 representation\_item  
 {representation\_item  
 representation\_item.name = 'draft'}  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.19.5.2 drafted\_surface to face\_shape\_element(as shape\_element)

AIM element: PATH  
Reference Path: drafted\_surface<=  
shape\_representation<=  
representation<=  
representation\_relationship.rep1  
representation\_relationship  
representation\_relationship.rep2  
representation=>  
shape\_representation=>  
face\_shape\_representation

### 5.1.19.5.3 drafted\_surface to planar\_element(as shape\_element)

AIM element: PATH  
Reference Path: drafted\_surface<=  
shape\_representation<=  
representation<=  
representation\_relationship.rep1  
representation\_relationship  
representation\_relationship.rep2  
representation=>  
shape\_representation=>  
planar\_shape\_representation

### 5.1.19.6 Edge\_shape\_element

AIM element: edge\_shape\_representation  
Source: ISO 10303-223  
Reference Path: edge\_shape\_representation <=  
shape\_representation<=  
representation  
representation.items[i] ->  
representation\_item =>  
geometric\_representation\_item =>  
(edge\_curve)  
(oriented\_edge)

### 5.1.19.7 Face\_shape\_element

AIM element: face\_shape\_representation  
 Source: ISO 10303-522  
 Reference Path: face\_shape\_representation <=  
 shape\_representation <=  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 geometric\_representation\_item =>  
 (face\_surface)  
 (oriented\_face)

### 5.1.19.8 Face\_shape\_element\_relationship

AIM element: face\_shape\_representation\_relationship  
 Source: ISO 10303-522  
 Reference Path: face\_shape\_representation\_relationship <=  
 representation\_relationship

#### 5.1.19.8.1 face\_shape\_element\_relationship to face\_shape\_element (as predecessor)

AIM element: PATH  
 Reference Path: face\_shape\_representation\_relationship <=  
 representation\_relationship  
 representation\_relationship.rep\_1->  
 representation=>  
 shape\_representation=>  
 face\_shape\_representation

#### 5.1.19.8.2 face\_shape\_element\_relationship to face\_shape\_element (as successor)

Reference Path: face\_shape\_representation\_relationship <=  
 representation\_relationship  
 representation\_relationship.rep\_2->  
 representation=>  
 shape\_representation=>  
 face\_shape\_representation

ISO 10303-223:2008 (E)

### 5.1.19.9 Geometric\_model

#1: advanced\_boundary\_rep

AIM element: [manifold\_solid\_brep]  
Source: ISO 10303-42

#### 5.1.19.10 Location\_element

AIM element: location\_shape\_representation  
Source: ISO 10303-522  
Reference Path: location\_shape\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'orientation'}  
{representation\_item =>  
geometric\_representation\_item =>  
point=>  
cartesian\_point}

#### 5.1.19.11 Model\_element

AIM element: (geometric\_representation\_item)  
(topological\_representation\_item)  
Source: ISO 10303-42

##### 5.1.19.11.1 model\_element to geometric\_model (as element)

#1: advanced\_boundary\_rep

AIM element: PATH  
Reference Path: [(geometric\_representation\_item =>  
surface <-  
face\_surface.face\_geometry  
face\_surface <=  
face <-)  
(topological\_representation\_item =>  
face <-)  
connected\_face\_set.cfs\_faces[i]  
connected\_face\_set =>  
closed\_shell <-  
manifold\_solid\_brep.outer  
manifold\_solid\_brep]

### 5.1.19.11.2 **model\_element to shape\_representation\_type (as representation\_type)**

AIM element: PATH  
 Reference Path: geometric\_representation\_item <=  
                   representation\_item <=  
                   representation.items[i]  
                   representation=>  
                   shape\_representation

### 5.1.19.12 **Object\_element\_shape\_representation**

AIM element: shape\_representation  
 Source: ISO 10303-41

#### 5.1.19.12.1 **object\_element\_shape\_representation to shape\_representation\_type (as representation\_type)**

AIM element: IDENTICAL MAPPING  
 Source:

#### 5.1.19.12.2 **object\_element\_shape\_representation to geometric\_model (as shape\_definition)**

#1: advanced\_boundary\_rep

AIM element: PATH  
 Reference Path: shape\_representation <=  
                   representation  
                   representation.items[i] ->  
                   representation\_item =>  
                   geometric\_representation\_item =>  
                   [solid\_model =>  
                   manifold\_solid\_brep]

### 5.1.19.13 **Orientation**

AIM element: placement  
 Source: ISO 10303-42

ISO 10303-223:2008 (E)

### **5.1.19.13.1 axis**

AIM element: (axis1\_placement.axis)  
((axis2\_placement\_3d.axis)  
(axis2\_placement\_3d.ref\_direction))  
Source: ISO 10303-42  
Reference Path: placement =>  
(axis1\_placement  
axis1\_placement.axis)  
(axis2\_placement\_3d  
(axis2\_placement\_3d.axis)  
(axis2\_placement\_3d.ref\_direction))

### **5.1.19.13.2 location**

AIM element: placement.location  
Source: ISO 10303-42

### **5.1.19.14 Part\_shape**

AIM element: product\_definition\_shape  
Source: ISO 10303-41

### 5.1.19.14.1 part\_shape to base\_shape (as base\_shape\_definition)

AIM element: PATH  
 Source: ISO 10303-41  
 Rule: material\_is\_specified\_for\_part (see 5.2.4.19)  
 Reference path: product\_definition\_shape <=  
   property\_definition  
   property\_definition.definition ->  
   characterized\_definition  
   characterized\_definition = characterized\_product\_definition  
   characterized\_product\_definition  
   characterized\_product\_definition = product\_definition  
   product\_definition <=  
   product\_definition\_relationship.relying\_product\_definition  
   {product\_definition\_relationship =>  
   product\_definition\_usage =>  
   make\_from\_usage\_option}  
   product\_definition\_relationship  
   product\_definition\_relationship.related\_product\_definition ->  
   product\_definition  
   characterized\_product\_definition = product\_definition  
   characterized\_product\_definition  
   characterized\_definition = characterized\_product\_definition  
   characterized\_definition <=  
   property\_definition.definition  
   property\_definition =>  
   product\_definition\_shape

### 5.1.19.14.2 part\_shape to object\_element\_shape\_representation (as representation\_form)

AIM element: PATH  
 Rules:  
 Reference Path: product\_definition\_shape <=  
   property\_definition <=  
   property\_definition\_representation.definition  
   {property\_definition\_representation =>  
   shape\_definition\_representation}  
   property\_definition\_representation  
   property\_definition\_representation.used\_representation ->  
   representation =>  
   shape\_representation =>  
   advanced\_brep\_shape\_representation

### 5.1.19.14.3 part\_shape to shape\_aspect (as element)

AIM element: PATH  
Reference Path: product\_definition\_shape <-  
shape\_aspect.of\_shape  
shape\_aspect

### 5.1.19.15 Parting\_surface

AIM element: parting\_surface  
Source: ISO 10303-223  
Reference Path: parting\_surface<=  
shape\_representation<=  
representation

#### 5.1.19.15.1 cope\_or\_drag

AIM element: PATH  
Reference Path: parting\_surface<=  
shape\_representation<=  
representation  
representation.items[i] ->  
representation\_item  
{representation\_item  
representation\_item.name = 'cope or drag'}  
descriptive\_representation\_item  
descriptive\_representation\_item = 'cope'  
descriptive\_representation\_item = 'drag'

#### 5.1.19.15.2 male\_or\_female

AIM element: PATH  
Reference Path: parting\_surface<=  
shape\_representation<=  
representation  
representation.items[i] ->  
representation\_item  
{representation\_item  
representation\_item.name = 'male or female'}  
descriptive\_representation\_item  
descriptive\_representation\_item = 'male'  
descriptive\_representation\_item = 'female'



**5.1.19.15.3 parting\_surface to numeric\_parameter (as offset\_distance)**

AIM element: PATH  
 Reference Path: parting\_surface<=  
 shape\_representation<=  
 representation  
 representation.items[i] ->  
 representation\_item  
 {representation\_item  
 representation\_item.name = 'offset distance'}  
 measure\_representation\_item

**5.1.19.15.4 parting\_surface to face\_shape\_element(as shape\_element)**

AIM element: PATH  
 Reference Path: parting\_surface<=  
 shape\_representation<=  
 representation<-  
 representation\_relationship.rep1  
 representation\_relationship  
 representation\_relationship.rep2  
 representation=>  
 shape\_representation=>  
 face\_shape\_representation

**5.1.19.15.5 parting\_surface to planar\_element(as shape\_element)**

AIM element: PATH  
 Reference Path: parting\_surface<=  
 shape\_representation<=  
 representation<-  
 representation\_relationship.rep1  
 representation\_relationship  
 representation\_relationship.rep2  
 representation=>  
 shape\_representation=>  
 planar\_shape\_representation

### 5.1.19.16 Path\_element

AIM element: path\_shape\_representation  
Source: ISO 10303-522  
Reference Path: path\_shape\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
{representation\_item  
representation\_item.name = 'profile shape'}  
representation\_item =>  
(geometric\_representation\_item =>  
curve=>  
bounded\_curve)  
(topological\_representation\_item=>  
edge=>  
edge\_curve)

### 5.1.19.17 Planar\_element

AIM element: planar\_shape\_representation  
Source: ISO 10303-522  
Reference Path: planar\_shape\_representation <=  
shape\_representation <=  
representation

#### 5.1.19.17.1 location

AIM element: cartesian\_point  
Source: ISO 10303-42  
Reference Path: planar\_shape\_representation <=  
shape\_representation <=  
representation  
representation.items[i] ->  
representation\_item =>  
geometric\_representation\_item =>  
surface=>  
elementary\_surface  
{elementary\_surface=>  
plane}  
elementary\_surface.position->  
axis2\_placement\_3d <=  
placement  
placement.location->  
cartesian\_point

**5.1.19.17.2 normal**

AIM element: direction  
 Source: ISO 10303-42  
 Reference Path: planar\_shape\_representation<=  
 shape\_representation<=  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 geometric\_representation\_item =>  
 surface=>  
 elementary\_surface  
 {elementary\_surface=>  
 plane}  
 elementary\_surface.position->  
 axis2\_placement\_3d.axis->  
 direction

**5.1.19.18 Shape\_aspect**

AIM element: shape\_aspect  
 Source: ISO 10303-41

**5.1.19.18.1 shape\_aspect to model\_element (as representation\_shape)**

AIM element: PATH  
 Reference Path: shape\_aspect  
 shape\_definition = shape\_aspect  
 shape\_definition  
 characterized\_definition = shape\_definition  
 characterized\_definition <-  
 property\_definition.definition  
 property\_definition <-  
 property\_definition\_representation.definition  
 property\_definition\_representation  
 property\_definition\_representation.used\_representation ->  
 representation  
 representation.items[i] ->  
 representation\_item =>  
 (geometric\_representation\_item)  
 (topological\_representation\_item)

### 5.1.19.18.2 shape\_aspect to shape\_aspect\_representation (as representation\_form)

AIM element: PATH  
Reference Path: shape\_aspect  
                  shape\_definition = shape\_aspect  
                  shape\_definition  
                  characterized\_definition = shape\_definition  
                  characterized\_definition <-  
                  property\_definition.definition  
                  property\_definition <-  
                  property\_definition\_representation.definition  
                  property\_definition\_representation  
                  property\_definition\_representation.used\_representation ->  
                  representation =>  
                  shape\_representation =>  
                  advanced\_brep\_shape\_representation

### 5.1.19.18.3 shape\_aspect to shape\_element (as element)

AIM element: IDENTICAL MAPPING

### 5.1.19.19 Shape\_aspect\_representation

AIM element: #1:[advanced\_brep\_shape\_representation]  
Source: #1: ISO 10303-514

#### 5.1.19.19.1 shape\_aspect\_representation to geometric\_model (as shape\_definition)

#1: advanced\_boundary\_rep

AIM element: PATH  
Reference Path: [advanced\_brep\_shape\_representation]  
                  [solid\_model =>  
                  manifold\_solid\_brep]

#### 5.1.19.20 Shape\_element

AIM element: shape\_aspect  
Source: ISO 10303-41  
Reference Path: {shape\_aspect  
                  shape\_aspect.product\_definitional = 'TRUE'}

### 5.1.19.21 Shape\_representation\_type

AIM element: shape\_representation  
Source: ISO 10303-41

#### 5.1.19.21.1 geometry\_type

#1: advanced\_boundary\_rep

AIM element: [advanced\_brep\_shape\_representation]  
Source: ISO 10303-514  
Reference Path: #1: [advanced\_brep\_shape\_representation <=  
shape\_representation ]

### 5.1.20 Simulation UoF

#### 5.1.20.1 Boundary\_condition

AIM element: boundary\_condition  
Source: ISO 10303-223  
Reference Path: boundary\_condition<=  
action\_method

##### 5.1.20.1.1 boundary\_condition to numeric\_parameter(as condition\_value)

AIM element: PATH  
Reference Path: boundary\_condition<=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation  
representation.name='condition value'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.20.1.2 boundary\_condition to property\_relationship(as condition\_value)

AIM element: PATH  
Reference Path: boundary\_condition<=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
{representation  
representation.name='condition value'}  
representation=>  
property\_relationship

### 5.1.20.1.3 boundary\_condition to shape\_aspect (as applies\_to)

AIM element: PATH  
Reference Path: boundary\_condition<=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=shape\_aspect  
shape\_aspect

## 5.1.20.2 Condition\_or\_assumption

AIM element: condition\_or\_assumption  
Source: ISO 10303-223  
Reference Path: condition\_or\_assumption<=  
action\_method

### 5.1.20.2.1 description

AIM element: resource\_property.description  
Source: ISO 10303-223  
Reference Path: condition\_or\_assumption<=  
action\_method  
action\_method.description

### 5.1.20.2.2 **condition\_or\_assumption to numeric\_parameter(as associated \_ - property)**

AIM element: PATH  
 Reference Path: condition\_or\_assumption<=  
 action\_method<-  
 action\_method\_assignment.assigned\_action  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definiton  
 property\_definiton<-  
 property\_definiton\_representation  
 property\_definiton\_representation.used\_representation->  
 representation  
 representation.name='associated property'  
 representation.items [i]->  
 representation\_item =>  
 measure\_representation\_item  
 {measure\_representation\_item <=  
 measure\_with\_unit =>  
 length\_measure\_with\_unit}

### 5.1.20.2.3 **condition\_or\_assumption to property\_relationship(as associated \_ - property)**

AIM element: PATH  
 Reference Path: condition\_or\_assumption <=  
 action\_method<-  
 action\_method\_assignment.assigned\_action  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definiton  
 property\_definiton<-  
 property\_definiton\_representation  
 property\_definiton\_representation.used\_representation->  
 {representation  
 representation.name='associated property'}  
 representation=>  
 property\_relationship

#### **5.1.20.2.4 condition\_or\_assumption to document\_assignment (as further\_information)**

AIM element: local\_time  
Source: ISO 10303-41  
Reference Path: condition\_or\_assumption<=  
action\_method  
document\_reference\_item=action\_method  
document\_reference\_item<=  
applied\_document\_reference.items [i]  
applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document

#### **5.1.20.3 Defect\_prediction**

AIM element: defect\_prediction  
Source: ISO 10303-226  
Reference Path: defect\_prediction<=  
action\_method

##### **5.1.20.3.1 description**

AIM element: action\_property.description  
Source: ISO 10303-49  
Reference Path: defect\_prediction<=  
action\_method  
action\_methoddescription

##### **5.1.20.3.2 type\_of\_defect**

AIM element: action\_property.name  
Source: ISO 10303-49  
Reference Path: defect\_prediction<=  
action\_method  
action\_method.name



### 5.1.20.3.3 defect\_prediction to location\_element(as location)

AIM element: PATH  
 Reference Path: defect\_prediction <=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definiton  
                   property\_definiton<-  
                   property\_definiton\_representation  
                   property\_definiton\_representation.used\_representation->  
                   representation=>  
                   shape\_representation=>  
                   location\_shape\_representation

### 5.1.20.3.4 defect\_prediction to shape\_aspect(as defect\_shape)

AIM element: PATH  
 Reference Path: defect\_prediction <=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=shape\_aspect  
                   shape\_aspect

### 5.1.20.4 Integration\_interval

AIM element: integration\_interval  
 Source: ISO 10303-223  
 Reference Path: integration\_interval<=  
                   action\_method

#### 5.1.20.4.1 integration\_interval to numeric\_parameter(as delta\_time)

AIM element: PATH  
Reference Path: integration\_interval<=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation  
representation.name='delta time'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

#### 5.1.20.4.2 integration\_interval to numeric\_parameter(as length\_of\_elapsed\_time\_interval)

AIM element: PATH  
Reference Path: integration\_interval<=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation  
representation.name='length of elapsed time interval'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.20.5 Meshing\_condition

AIM element: meshing\_condition  
 Source: ISO 10303-223  
 Reference Path: meshing\_condition<=  
 action\_method

#### 5.1.20.5.1 number\_of\_elements

AIM element: count\_measure  
 Source: ISO 10303-41  
 Reference Path: meshing\_condition <=  
 action\_method<-  
 action\_method\_assignment.assigned\_action  
 action\_method\_assignment=>  
 applied\_action\_method\_assignment  
 applied\_action\_method\_assignment.items[i]->  
 action\_method\_item  
 action\_method\_item=property\_definiton  
 property\_definiton<-  
 property\_definiton\_representation  
 property\_definiton\_representation.used\_representation->  
 representation  
 representation.name='number of elements'  
 representation.items [i]->  
 representation\_item =>  
 measure\_representation\_item  
 measure\_with\_unit  
 measure\_with\_unit.value\_component ->  
 measure\_value  
 measure\_value = count\_measure  
 count\_measure

### 5.1.20.5.2 number\_of\_nodes

AIM element: count\_measure  
Source: ISO 10303-41  
Reference Path: meshing\_condition <=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation  
representation.name='number of nodes'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
measure\_with\_unit  
measure\_with\_unit.value\_component ->  
measure\_value  
measure\_value = count\_measure  
count\_measure

### 5.1.20.5.3 meshing\_condition to numeric\_range(as density\_of\_nodes)

AIM element: PATH  
 Reference Path: meshing\_condition <=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definiton  
                   property\_definiton<-  
                   property\_definiton\_representation  
                   property\_definiton\_representation.used\_representation->  
                   representation  
                   representation.name='density of nodes'  
                   representation.items [i]->  
                   representation\_item =>  
                   measure\_representation\_item  
                   {measure\_representation\_item <=  
                   measure\_with\_unit =>  
                   length\_measure\_with\_unit}

### 5.1.20.5.4 meshing\_condition to numeric\_range(as density\_of\_elements)

AIM element: PATH  
 Reference Path: meshing\_condition <=  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=property\_definiton  
                   property\_definiton<-  
                   property\_definiton\_representation  
                   property\_definiton\_representation.used\_representation->  
                   representation  
                   representation.name='density of elements'  
                   representation.items [i]->  
                   representation\_item =>  
                   measure\_representation\_item  
                   {measure\_representation\_item <=  
                   measure\_with\_unit =>  
                   length\_measure\_with\_unit}

### 5.1.20.5 meshing\_condition to shape\_aspect (as area\_to\_be\_meshed)

AIM element: PATH  
Reference Path: meshing\_condition <=  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=shape\_aspect  
shape\_aspect

### 5.1.20.6 Simulation\_input

AIM element: simulation\_input  
Source: ISO 10303-223  
Reference Path: simulation\_input<=  
requirement\_for\_action\_resource

#### 5.1.20.6.1 simulation\_input to document\_assignment(as setup\_file)

AIM element: local\_time  
Source: ISO 10303-41  
Reference Path: simulation\_input<=  
requirement\_for\_action\_resource  
document\_reference\_item=requirement\_for\_action\_resource  
document\_reference\_item<-  
applied\_document\_reference.items [i]  
applied\_document\_reference<=  
document\_reference  
document\_reference.assigned\_document->  
document

#### 5.1.20.6.2 simulation\_input to condition\_or\_assumption(as condition)

AIM element: PATH  
Reference Path: simulation\_input<=  
requirement\_for\_action\_resource<=  
action\_resource\_requirement  
action\_resource\_requirement.operations[i]->  
characterized\_action\_definition  
characterized\_action\_definition=action\_method  
action\_method=>  
(action\_method.name='condition')  
condition\_or\_assumption

**5.1.20.6.3 simulation\_input to condition\_or\_assumption(as assumption)**

AIM element: PATH  
 Reference Path: simulation\_input<=  
 requirement\_for\_action\_resource<=  
 action\_resource\_requirement  
 action\_resource\_requirement.operations[i]->  
 characterized\_action\_definition  
 characterized\_action\_definition=action\_method  
 action\_method=>  
 (action\_method.name='assumption')  
 condition\_or\_assumption

**5.1.20.6.4 simulation\_input to integration\_interval(as integration\_time\_specification)**

AIM element: PATH  
 Reference Path: simulation\_input<=  
 requirement\_for\_action\_resource<=  
 action\_resource\_requirement  
 action\_resource\_requirement.operations[i]->  
 characterized\_action\_definition  
 characterized\_action\_definition=action\_method  
 action\_method=>  
 (action\_method.name='integration time specification')  
 integration\_interval

### 5.1.20.6.5 simulation\_input to numeric\_parameter(as input\_parameter)

AIM element: PATH  
Reference Path: simulation\_input<=  
requirement\_for\_action\_resource<=  
action\_resource\_requirement  
characterized\_resource\_definition=action\_resource\_requirement  
characterized\_resource\_definition<=  
resource\_property.resource  
resource\_property<=  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.name='input parameter'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

### 5.1.20.6.6 simulation\_input to property\_relationship(as input\_parameter)

AIM element: PATH  
Reference Path: simulation\_input<=  
requirement\_for\_action\_resource<=  
action\_resource\_requirement  
characterized\_resource\_definition=action\_resource\_requirement  
characterized\_resource\_definition<=  
resource\_property.resource  
resource\_property<=  
resource\_property\_representation.property  
resource\_property\_representation  
resource\_property\_representation.representation ->  
representation  
representation.name='input parameter'  
representation.items [i]->  
representation\_item =>  
{representation.name='input parameter'}  
property\_relationship



### 5.1.20.6.7 simulation\_input to simulation\_input\_region (as region)

AIM element: PATH  
 Reference Path: simulation\_input<=  
                   requirement\_for\_action\_resource  
                   requirement\_for\_action\_resource.resources[i]->  
                   action\_resource=>  
                   simulation\_input\_region

### 5.1.20.7 Simulation\_input\_region

AIM element: simulation\_input\_region  
 Source: ISO 10303-223  
 Reference Path: simulation\_input\_region<=  
                   action\_resource

#### 5.1.20.7.1 region\_type

AIM element: action\_resource\_type.name  
 Source: ISO 10303-41  
 Reference Path: simulation\_input\_region<=  
                   action\_resource  
                   action\_resource.kind->  
                   action\_resource\_type  
                   action\_resource\_type.name

#### 5.1.20.7.2 simulation\_input\_region to boundary\_conditon(as boundary\_ - condition)

AIM element: PATH  
 Reference Path: simulation\_input\_region<=  
                   action\_resource  
                   action\_resource.usage[i]->  
                   action\_method=>  
                   boundary\_condition

#### 5.1.20.7.3 simulation\_input\_region to meshing\_conditon(as meshing\_ - condition)

AIM element: PATH  
 Reference Path: simulation\_input\_region<=  
                   action\_resource  
                   action\_resource.usage[i]->  
                   action\_method=>  
                   meshing\_condition

#### 5.1.20.7.4 simulation\_input\_region to numeric\_parameter(as region\_property)

AIM element: PATH  
Reference Path: simulation\_input\_region<=  
action\_resource  
action\_resource.usage[i]->  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation  
representation.name='region property'  
representation.items [i]->  
representation\_item =>  
measure\_representation\_item  
{measure\_representation\_item <=  
measure\_with\_unit =>  
length\_measure\_with\_unit}

#### 5.1.20.7.5 simulation\_input\_region to property\_relationship(as region\_property)

AIM element: PATH  
Reference Path: simulation\_input\_region<=  
action\_resource  
action\_resource.usage[i]->  
action\_method<-  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definiton  
property\_definiton<-  
property\_definiton\_representation  
property\_definiton\_representation.used\_representation->  
representation=>  
{representation.name='region property'}  
property\_relationship

### 5.1.20.7.6 simulation\_input\_region to shape\_element (as simulation\_region\_shape)

AIM element: PATH  
 Reference Path: simulation\_input\_region<=  
                   action\_resource  
                   action\_resource.usage[i]->  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=shape\_aspect  
                   shape\_aspect

### 5.1.20.7.7 simulation\_input\_region to part\_version (as associated\_part)

AIM element: PATH  
 Reference Path: simulation\_input\_region<=  
                   action\_resource  
                   action\_resource.usage[i]->  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=product\_definition  
                   product\_definition

### 5.1.20.8 Simulation\_output

AIM element: simulation\_output  
 Source: ISO 10303-226  
 Reference Path: simulation\_output<=  
                   versioned\_action\_request

### 5.1.20.8.1 simulation\_output to defect\_prediction(as predicted\_defect)

AIM element: PATH  
Reference Path: simulation\_output<=  
versioned\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method=>  
defect\_prediction

### 5.1.20.8.2 simulation\_output to simulation\_result(as result)

AIM element: PATH  
Reference Path: simulation\_output<=  
versioned\_action\_request<-  
action\_request\_solution.request  
action\_request\_solution  
action\_request\_solution.method->  
action\_method=>  
simulation\_result

### 5.1.20.9 Simulation\_output\_region

AIM element: simulation\_output\_region  
Source: ISO 10303-226  
Reference Path: simulation\_output\_region<=  
action\_property

#### 5.1.20.9.1 simulation\_output\_region to simulation\_input\_region(as belongs\_to)

AIM element: PATH  
Reference Path: simulation\_output\_region<=  
action\_property  
action\_property.definition->  
characterized\_action\_definition  
characterized\_action\_definition=action\_method  
action\_method<-  
supported\_item=action\_method  
supported\_item<-  
action\_resource.usage[i]  
action\_resource=>  
simulation\_input\_region

**5.1.20.9.2 simulation\_output\_region to simulation\_unit\_state(as history)**

AIM element: PATH  
 Reference Path: simulation\_output\_region<=  
                   action\_property<-  
                   action\_property\_relationship.relating\_action\_property  
                   action\_property\_relationship  
                   action\_property\_relationship.related\_action\_property->  
                   action\_property=>  
                   simulation\_unit\_state

**5.1.20.9.3 simulation\_output\_region to shape\_element(as region\_shape)**

AIM element: PATH  
 Reference Path: simulation\_output\_region<=  
                   action\_property<-  
                   action\_property.definition->  
                   characterized\_action\_definition  
                   characterized\_action\_definition=action\_method  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=shape\_aspect  
                   shape\_aspect

**5.1.20.10 Simulated\_property**

AIM element: simulated\_property  
 Source: ISO 10303-226  
 Reference Path: simulated\_property<=  
                   representation

**5.1.20.10.1 name**

AIM element: representation.name  
 Source: ISO 10303-41  
 Reference Path: simulated\_property<=  
                   representation  
                   representation.name

### 5.1.20.10.2 property\_value

AIM element: representation  
Source: ISO 10303-41  
Reference Path: simulation\_property<=  
representation  
representation.items[i]->  
representation\_item=>  
(representation\_item.name='property value'  
measure\_representation\_item<=  
measure\_with\_unit  
measure\_with\_unit.value\_component  
parameter\_value

### 5.1.20.11 Simulation\_result

AIM element: simulation\_result  
Source: ISO 10303-223  
Reference Path: simulation\_result<=  
action\_request\_solution

#### 5.1.20.11.1 begin\_time

AIM element: local\_time  
Source: ISO 10303-41  
Reference Path: simulation\_result  
time\_item= simulation\_unit\_state  
time\_item<-  
applied\_time\_assignment.items[i]  
applied\_time\_assignment<=  
time\_assignment  
(time\_assignment.role->  
time\_role  
time\_role.name= 'begin time'  
time\_assignment.assigned\_time->  
local\_time

**5.1.20.11.2 end\_time**

AIM element: local\_time  
 Source: ISO 10303-41  
 Reference Path: simulation\_result  
                   time\_item= simulation\_unit\_state  
                   time\_item<-  
                   applied\_time\_assignment.items[i]  
                   applied\_time\_assignment<=  
                   time\_assignment  
                   (time\_assignment.role->  
                   time\_role  
                   time\_role.name= 'end time'  
                   time\_assignment.assigned\_time->  
                   local\_time

**5.1.20.11.3 simulation\_result to simulation\_output\_region (as region)**

AIM element: PATH  
 Reference Path: simulation\_result<=  
                   action\_request\_solution  
                   action\_request\_solution.method  
                   action\_method  
                   action\_method\_relationship.relating\_action\_method  
                   action\_method\_relationship  
                   action\_method\_relationship.related\_action\_method->  
                   action\_method<-  
                   characterized\_action\_definition=action\_method  
                   characterized\_action\_definition<-  
                   action\_property.definition  
                   action\_property=>  
                   simulation\_output\_region

**5.1.20.11.4 simulation\_result to illustration (as illustration)**

AIM element: PATH  
 Reference Path: simulation\_result  
                   document\_reference\_item= simulation\_result  
                   document\_reference\_item <-  
                   applied\_document\_reference.items [i]  
                   applied\_document\_reference <=  
                   document\_reference  
                   document\_reference.assigned\_document ->  
                   document=>  
                   illustration

### 5.1.20.12 Simulation\_run

AIM element: simulation\_run  
Source: ISO 10303-226  
Reference Path: simulation\_run<=  
directed\_action<=  
executed\_action<=  
action

#### 5.1.20.12.1 description

AIM element: action.description  
Source: ISO 10303-41  
Reference Path: simulation\_run<=  
directed\_action<=  
executed\_action<=  
action  
action.description

#### 5.1.20.12.2 identification

AIM element: id\_attribute.attribute\_value  
Source: ISO 10303-41  
Reference Path: simulation\_run<=  
directed\_action<=  
executed\_action<=  
action  
id\_attribute\_select=action  
id\_attribute\_select<-  
id\_attribute.identified\_item  
id\_attribute  
id\_attribute.attribute\_value

#### 5.1.20.12.3 simulation\_run to activity (as associated\_process)

AIM element: PATH  
Reference Path: simulation\_run<=  
executed\_action<=  
action  
action.chosen\_method->  
action\_method=>  
process\_plan\_activity



**5.1.20.12.4 simulation\_run to simulation\_input (as input)**

AIM element: PATH  
 Reference Path: simulation\_run<=  
                   executed\_action  
                   action  
                   characterized\_action\_definition=action  
                   characterized\_action\_definition<-  
                   action\_resource\_requirements.operations[i]  
                   action\_resource\_requirements=>  
                   requirement\_for\_action\_resource=>  
                   simulation\_input

**5.1.20.12.5 simulation\_run to simulation\_output (as output)**

AIM element: PATH  
 Reference Path: simulation\_run<=  
                   executed\_action=>  
                   directed\_action  
                   direction\_action.directive->  
                   action\_directive  
                   action\_directive.requests[i]->  
                   version\_action\_request=>  
                   simulation\_output

**5.1.20.12.6 simulation\_run to simulation\_software (as software\_used)**

AIM element: PATH  
 Reference Path: simulation\_run<=  
                   executed\_action<=  
                   action  
                   supported\_item=action  
                   supported\_item<-  
                   action\_resource.usage[i]  
                   action\_resource=>  
                   simulation\_software

**5.1.20.13 Simulation\_run\_relationship**

AIM element: simulation\_run\_relationship  
 Source: ISO 10303-223  
 Reference Path: simulation\_run\_relationship<=  
                   action\_method\_relationship

### 5.1.20.13.1 description

AIM element: action\_method\_relationship.description  
Source: ISO 10303-41  
Reference Path: simulation\_run\_relationship<=  
action\_method\_relationship  
action\_method\_relationship.description

### 5.1.20.13.2 simulation\_run\_relationship to simulation\_run (as related)

AIM element: PATH  
Reference Path: simulation\_run\_relationship<=  
action\_method\_relationship  
action\_method\_relationship.related\_method->  
action\_method=>  
simulation\_run

### 5.1.20.13.3 simulation\_run\_relationship to simulation\_run (as relating)

AIM element: PATH  
Reference Path: simulation\_run\_relationship<=  
action\_method\_relationship  
action\_method\_relationship.relatng\_method->  
action\_method=>  
simulation\_run

### 5.1.20.14 Simulation\_software

AIM element: simulation\_software  
Source: ISO 10303-226  
Reference Path: simulation\_software<=  
action\_resource

### 5.1.20.14.1 simulation\_software to date (as release\_date)

AIM element: calendar\_date  
Source: ISO 10303-41  
Rules: dependent\_instantiable\_date – (see 5.2.4.8 )  
Reference Path: simulation\_software<=  
action\_resource

```

date_item = action_resource
date_item <-
applied_date_assignment.items [i]
applied_date_assignment <=
date_assignment
{date_assignment.role ->
date_role
date_role.name='release date'}
date_assignment.assigned_date ->
date

```

### 5.1.20.14.2 software\_name

AIM element: action\_resource.name  
 Source: ISO 10303-41  
 Reference Path: simulation\_software<=  
 action\_resource  
 action\_resource.name

### 5.1.20.14.3 software\_version

AIM element: action\_resource.description  
 Source: ISO 10303-41  
 Reference Path: simulation\_software<=  
 action\_resource  
 action\_resource.description

### 5.1.20.15 Simulation\_unit

AIM element: simulation\_unit  
 Source: ISO 10303-226  
 Reference Path: simulation\_unit<=  
 action\_property

#### 5.1.20.15.1 name

AIM element: action\_property\_representation.name  
 Source: ISO 10303-41  
 Reference Path: simulation\_unit<=  
 action\_property  
 action\_property.name

### 5.1.20.15.2 simulation\_unit to location\_element(as unit\_location)

AIM element: PATH  
Reference Path: simulation\_unit <=  
action\_property <=  
action\_property.definition->  
characterized\_action\_definition  
characterized\_action\_definition=action\_method  
action\_method <=  
action\_method\_assignment.assigned\_action  
action\_method\_assignment=>  
applied\_action\_method\_assignment  
applied\_action\_method\_assignment.items[i]->  
action\_method\_item  
action\_method\_item=property\_definition  
property\_definition <=  
property\_definition\_representation  
property\_definition\_representation.used\_representation->  
representation=>  
shape\_representation=>  
location\_shape\_representation

### 5.1.20.15.3 simulation\_unit to simulation\_output\_region(as belongs\_to)

AIM element: PATH  
Reference Path: simulation\_unit <=  
action\_property <=  
action\_property\_relationship.relying\_action\_property  
action\_property\_relationship  
action\_property\_relationship.related\_action\_property->  
action\_property=>  
simulation\_output\_region

### 5.1.20.15.4 simulation\_unit to simulation\_unit\_state(as history)

AIM element: PATH  
Reference Path: simulation\_unit <=  
action\_property <=  
action\_property\_relationship.relying\_action\_property  
action\_property\_relationship  
action\_property\_relationship.related\_action\_property->  
action\_property=>  
simulation\_unit\_state

**5.1.20.15.5 simulation\_unit to shape\_aspect(as unit\_shape)**

AIM element: PATH  
 Reference Path: simulation\_unit<=  
                   action\_property<-  
                   action\_property.definition->  
                   characterized\_action\_definition  
                   characterized\_action\_definition=action\_method  
                   action\_method<-  
                   action\_method\_assignment.assigned\_action  
                   action\_method\_assignment=>  
                   applied\_action\_method\_assignment  
                   applied\_action\_method\_assignment.items[i]->  
                   action\_method\_item  
                   action\_method\_item=shape\_aspect  
                   shape\_aspect

**5.1.20.16 Simulation\_unit\_state**

AIM element: simulation\_unit\_state  
 Source: ISO 10303-226  
 Reference Path: simulation\_unit\_state<=  
                   action\_property

**5.1.20.16.1 evaluation\_time**

AIM element: local\_time  
 Source: ISO 10303-41  
 Reference Path: simulation\_unit\_state<=  
                   action\_property  
                   time\_item= action\_property  
                   time\_item<-  
                   applied\_time\_assignment.items[i]  
                   applied\_time\_assignment<=  
                   time\_assignment  
                   (time\_assignment.role->  
                   time\_role  
                   time\_role.name= 'evaluation time'  
                   time\_assignment.assigned\_time->  
                   local\_time

### 5.1.20.16.2 simulation\_unit\_state to simulated\_property (as property)

AIM element: time\_measure\_with\_unit  
 Source: ISO 10303-41  
 Reference Path: simulation\_unit\_state<=  
                   action\_property\_representation  
                   action\_property\_representation.representation ->  
                   representation=>  
                   simulated\_property

## 5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and the AICs and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303. This clause also specifies modifications to the textual material for constructs that are imported from the integrated resources and the AICs. The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM. Requirements stated in the integrated resources which refer to such items and subtypes apply exclusively to those items which are imported into the AIM.

\*)

```

SCHEMA cast_parts_schema;

USE FROM aic_geometric_tolerances;           -- ISO 10303-519
    USE FROM aic_advanced_brep               -- ISO 10303-514
        (advanced_brep_shape_representation);

USE FROM aic_machining_feature;             -- ISO 10303-522

USE FROM aic_topologically_bounded_surface  -- ISO 10303-511
    (advanced_face);

USE FROM action_schema                      -- ISO 10303-41
    (action,
     action_directive,
     action_method,
     action_method_relationship,
     action_relationship,
     action_request_solution,
     action_resource,
     action_resource_relationship,
     action_resource_type,
     action_status,
     directed_action,
     executed_action,
     supported_item,
     versioned_action_request);

USE FROM application_context_schema         -- ISO 10303-41
    (application_context,

```

```

    application_context_element,
    application_protocol_definition,
    product_definition_context);

USE FROM approval_schema -- ISO 10303-41
    (approval,
    approval_date_time,
    approval_person_organization,
    approval_role,
    approval_status);

USE FROM basic_attribute_schema -- ISO 10303-41
    (id_attribute,
    description_attribute,
    description_attribute_select,
    name_attribute,
    object_role);

USE FROM date_time_schema -- ISO 10303-41
    (calendar_date,
    coordinated_universal_time_offset,
    date,
    date_and_time,
    date_role,
    date_time_select,
    local_time);

USE FROM document_schema -- ISO 10303-41
    (document,
    document_product_association,
    document_relationship,
    document_representation_type,
    document_type,
    document_usage_constraint,
    product_or_formation_or_definition);

USE FROM effectivity_schema -- ISO 10303-41
    (effectivity,
    dated_effectivity,
    lot_effectivity);

REFERENCE FROM geometry_schema -- ISO 10303-42
    (dummy_gri);

USE FROM external_reference_schema -- ISO 10303-41
    (external_source,
    externally_defined_item,
    externally_defined_item_relationship,
    pre_defined_item,
    source_item);

USE FROM geometry_schema -- ISO 10303-42
    (axis2_placement_2d,

```

ISO 10303-223:2008 (E)

```
axis2_placement_3d,  
bezier_curve,  
bezier_surface,  
b_spline_curve_with_knots,  
b_spline_surface_with_knots,  
cartesian_point,  
cartesian_transformation_operator_3d,  
circle,  
composite_curve_on_surface,  
conical_surface,  
cylindrical_surface,  
degenerate_toroidal_surface,  
direction,  
ellipse,  
geometric_representation_context,  
hyperbola,  
line,  
oriented_surface,  
parabola,  
pcurve,  
plane,  
point,  
polyline,  
quasi_uniform_curve,  
quasi_uniform_surface,  
rational_b_spline_curve,  
rational_b_spline_surface,  
spherical_surface,  
surface_curve,  
surface_of_linear_extrusion,  
surface_of_revolution,  
swept_surface,  
toroidal_surface,  
uniform_curve,  
uniform_surface,  
vector);
```

```
USE FROM geometric_model_schema -- ISO 10303-42  
  (faceted_brep,  
   manifold_solid_brep);
```

```
USE FROM group_schema -- ISO 10303-41  
  (group,  
   group_relationship);
```

```
USE FROM management_resources_schema -- ISO 10303-41  
  (action_assignment,  
   action_method_assignment,  
   action_method_role,  
   action_request_assignment,  
   approval_assignment,  
   classification_role,  
   classification_assignment,
```



```

date_assignment,
date_and_time_assignment,
document_reference,
document_usage_constraint_assignment,
effectivity_assignment,
external_identification_assignment,
group_assignment,
identification_assignment,
organization_assignment,
person_and_organization_assignment,
security_classification_assignment,
time_assignment);

USE FROM material_property_definition_schema           -- ISO 10303-45
(material_designation,
material_property,
product_material_composition_relationship,
property_definition_relationship);

USE FROM material_property_representation_schema       -- ISO 10303-45
(data_environment,
material_property_representation);

USE FROM measure_schema                               -- ISO 10303-41
(amount_of_substance_measure,
area_measure,
celsius_temperature_measure,
conversion_based_unit,
context_dependent_unit,
count_measure,
derived_unit,
electric_current_measure,
global_unit_assigned_context,
length_measure,
length_measure_with_unit,
length_unit,
luminous_intensity_measure,
mass_measure_with_unit,
mass_unit,
mass_measure,
measure_value,
measure_with_unit,
unit,
named_unit,
plane_angle_measure,
plane_angle_measure_with_unit,
plane_angle_unit,
positive_plane_angle_measure,
ratio_measure,
ratio_unit,
si_unit,
solid_angle_measure,
solid_angle_unit,

```

ISO 10303-223:2008 (E)

```
        time_measure,  
        time_measure_with_unit,  
        time_unit,  
        thermodynamic_temperature_measure,  
        volume_measure);  
  
USE FROM method_definition_schema -- ISO 10303-49  
    (action_method_with_associated_documents,  
     relationship_condition,  
     serial_action_method,  
     sequential_method);  
  
USE FROM person_organization_schema -- ISO 10303-41  
    (address,  
     organization,  
     organization_role,  
     organizational_address,  
     organizational_project,  
     person,  
     person_and_organization,  
     person_organization_select,  
     personal_address);  
  
USE FROM process_property_representation_schema -- ISO 10303-49  
    (action_property_representation,  
     resource_property_representation);  
  
USE FROM product_definition_schema -- ISO 10303-41  
    (product,  
     product_category,  
     product_definition,  
     product_definition_effectivity,  
     product_definition_with_associated_documents,  
     product_definition_formation,  
     product_definition_formation_relationship,  
     product_definition_relationship,  
     product_related_product_category);  
  
USE FROM process_property_schema -- ISO 10303-49  
    (action_property,  
     characterized_action_definition,  
     characterized_resource_definition,  
     action_resource_requirement,  
     action_property_relationship,  
     process_product_association,  
     process_property_association,  
     product_definition_process,  
     property_process,  
     requirement_for_action_resource,  
     resource_property);  
  
USE FROM product_property_definition_schema -- ISO 10303-41
```

```

(characterized_definition,
characterized_object,
characterized_product_definition,
general_property,
general_property_association,
product_definition_shape,
property_definition,
shape_definition,
shape_aspect,
shape_aspect_relationship);

REFERENCE FROM product_property_representation_schema      -- ISO 10303-41
  (get_property_definition_representations);

USE FROM product_property_representation_schema            -- ISO 10303-41
  (property_definition_representation,
  represented_definition,
  shape_definition_representation,
  shape_representation);

USE FROM product_structure_schema                          -- ISO 10303-41
  (assembly_component_usage,
  make_from_usage_option,
  next_assembly_usage_occurrence,
  product_definition_usage);

USE FROM qualified_measure_schema                          -- ISO 10303-45
  (descriptive_representation_item,
  measure_qualification,
  measure_representation_item,
  precision_qualifier,
  qualified_representation_item,
  standard_uncertainty,
  type_qualifier,
  uncertainty_qualifier,
  value_qualifier);

USE FROM representation_schema                             -- ISO 10303-43
  (compound_representation_item,
  global_uncertainty_assigned_context,
  representation_item,
  set_representation_item,
  parametric_representation_context,
  representation,
  representation_relationship,
  uncertainty_measure_with_unit,
  value_representation_item);

USE FROM security_classification_schema
  (security_classification,
  security_classification_level);

```

## ISO 10303-223:2008 (E)

```
USE FROM shape_dimension_schema                                -- ISO 10303-47
    (angular_location,
     angular_size,
     dimensional_characteristic_representation,
     dimensional_location,
     dimensional_location_with_path,
     dimensional_size,
     dimensional_size_with_path,
     shape_dimension_representation);

USE FROM shape_tolerance_schema                               -- ISO 10303-47
    (geometric_tolerance,
     limits_and_fits,
     plus_minus_tolerance,
     tolerance_value,
     tolerance_zone);

REFERENCE FROM support_resource_schema                        -- ISO 10303-41
    (bag_to_set,
     identifier);

USE FROM topology_schema                                     -- ISO 10303-42
    (closed_shell,
     connected_face_set,
     edge,
     edge_loop,
     face,
     face_surface,
     open_shell,
     poly_loop,
     shell,
     topological_representation_item);

REFERENCE FROM topology_schema                               -- ISO 10303-42
    (dummy_tri);
(*
```

NOTE The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
application_context_schema	ISO 10303-41
approval_schema	ISO 10303-41
aic_machining_feature	ISO 10303-522
aic_advanced_brep	ISO 10303-514
aic_geometric_tolerances	ISO 10303-519
aic_topologically_bounded_surface	ISO 10303-511
basic_attribute_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45
measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41

product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41
shape_aspect_definition_schema	ISO 10303-47
shape_dimension_schema	ISO 10303-47
shape_tolerance_schema	ISO 10303-47
support_resource_schema	ISO 10303-41
topology_schema	ISO 10303-42

## 5.2.1 Fundamental concepts and assumptions

## 5.2.2 Cast parts schema types

### 5.2.2.1 action\_item

An **action\_item** identifies an **action\_resource**, **characterized\_object**, **product\_definition**, **product\_definition\_formation** and **property\_definition** to which a referenced **action\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE action_item = SELECT (
    action_resource,
    characterized_object,
    product_definition,
    product_definition_formation,
    property_definition);
END_TYPE;
(*
```

### 5.2.2.2 action\_method\_item

An **action\_method\_item** identifies a **action\_method**, **shape\_aspect**, **externally\_defined\_item**, **product\_definition** and **property\_definition** to which a referenced **action\_method** may be assigned.

EXPRESS specification:

```
*)
TYPE action_method_item = SELECT
    (shape_aspect,
    externally_defined_item,
    product_definition,
    property_definition);
END_TYPE; -- action_method_item
(*
```

### 5.2.2.3 action\_request\_item

An **action\_request\_item** identifies a **document**, **product\_definition** and **property\_definition** to which a referenced **action\_request\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE action_request_item = SELECT (
    document,
    product_definition,
    property_definition);
END_TYPE;
(*
```

### 5.2.2.4 approved\_item

An **approval\_item** identifies a **action\_method**, **action\_relationship**, **directed\_action**, **product\_definition\_formation** and **versioned\_action\_request** to which a referenced **approval** may be assigned.

EXPRESS specification:

```
*)
TYPE approved_item = SELECT
    (action_method,
    action_relationship,
    directed_action,
    product_definition_formation,
    versioned_action_request);
END_TYPE; -- approved_item
(*
```

### 5.2.2.5 classification\_item

A **classification\_item** identifies a **externally\_defined\_representation\_with\_parameters** to which a referenced **classification\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE classification_item = SELECT (
    externally_defined_representation_with_parameters);
END_TYPE;
(*
```

### 5.2.2.6 date\_and\_time\_item

A **date\_and\_time\_item** identifies a **representation** to which a referenced **date\_and\_time\_assignment** may be assigned.

EXPRESS specification:

```

*)
TYPE date_and_time_item = SELECT
    (representation);
END_TYPE;
(*

```

**5.2.2.7 date\_item**

A **date\_item** identifies a **action**, **action\_resource**, **product\_definition**, **property\_definition**, **process\_plan\_security**, **document** and **versioned\_action\_request** to which a referenced **date** may be assigned.

EXPRESS specification:

```

*)
TYPE date_item = SELECT (
    action,
    action_resource,
    product_definition,
    property_definition,
    process_plan_security,
    document,
    versioned_action_request);
END_TYPE;
(*

```

**5.2.2.8 document\_reference\_item**

A **document\_reference\_item** identifies a **action**, **action\_method**, **action\_resource**, **action\_request\_solution**, **externally\_defined\_schema**, **externally\_defined\_feature\_definition**, **property\_definition**, **resource\_property**, **known\_source** and **requirement\_for\_action\_resource** to which a referenced **document\_reference** may be assigned.

EXPRESS specification:

```

*)
TYPE document_reference_item = SELECT (
    action,
    action_method,
    action_resource,
    action_request_solution,
    externally_defined_feature_definition,
    externally_defined_schema,
    property_definition,
    resource_property,
    requirement_for_action_resource,
    known_source);
END_TYPE;
(*

```

### 5.2.2.9 effectivity\_item

An **effectivity\_item** identifies a **property\_definition** and **product\_definition** to which a referenced **effectivity\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE effectivity_item = SELECT (
    product_definition,
    property_definition);
END_TYPE;
(*
```

### 5.2.2.10 external\_identification\_item

An **external\_identification\_item** identifies a **externally\_defined\_class**, **externally\_defined\_general\_property**, **product\_definition** and **document** to which a referenced **external\_identification\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE external_identification_item = SELECT (
    document,
    externally_defined_class,
    externally_defined_general_property,
    product_definition);
END_TYPE;
(*
```

### 5.2.2.11 group\_item

A **group\_item** identifies an **instanced\_feature**, **replicate\_feature** and **transition\_feature** to which a referenced **group\_assignment** may be assigned.

EXPRESS specification:

```
*)
TYPE group_item = SELECT
    (instanced_feature,
    replicate_feature,
    transition_feature);
END_TYPE;
(*
```

### 5.2.2.12 identification\_assignment\_item

An **identification\_assignment\_item** identifies a **document** and **document\_file** to which a referenced **identification\_assignment** may be assigned.



EXPRESS specification:

```

*)
TYPE identification_assignment_item= SELECT (
    document,
    document_file);
END_TYPE;
(*

```

**5.2.2.13 ordered\_item**

An **ordered\_item** identifies a **product\_definition\_formation** or **action** to which a referenced **ordered\_part** may be assigned.

EXPRESS specification:

```

*)
TYPE ordered_item = SELECT
    (action,product_definition_formation);
END_TYPE;
(*

```

**5.2.2.14 organization\_item**

An **organization\_item** identifies a **property\_definition**, **action\_method**, **action\_resource** and **known\_source** to which a referenced **organization\_assignment** may be assigned.

EXPRESS specification:

```

*)
TYPE organization_item = SELECT (
    action_method,
    action_resource,
    known_source,
    property_definition);
END_TYPE;
(*

```

**5.2.2.15 person\_and\_organization\_item**

A **person\_and\_organization\_item** identifies a **action**, **product\_definition\_formation**, **shape\_aspect**, **property\_definition** and **product\_definition** to which a referenced **person\_and\_organization\_assignment** may be assigned.

EXPRESS specification:

```
*)  
TYPE person_and_organization_item = SELECT (  
    action,  
    product_definition_formation,  
    product_definition,  
    property_definition,  
    shape_aspect,  
    versioned_action_request);  
END_TYPE;  
(*
```

### 5.2.2.16 security\_classification\_item

A **security\_classification\_item** identifies a **product\_definition\_formation**, **action** and **action\_method** to which a referenced **security\_classification\_assignment** may be assigned.

EXPRESS specification:

```
*)  
TYPE security_classification_item = SELECT  
    (action,  
    action_method,  
    product_definition_formation);  
END_TYPE;  
(*
```

### 5.2.2.17 time\_item

A **time\_item** identifies a **action\_property\_representation** and **action\_request\_solution** to which a referenced **time\_assignment** may be assigned.

EXPRESS specification:

```
*)  
TYPE time_item = SELECT  
    (action_property,  
    action_request_solution);  
END_TYPE;  
(*
```

## 5.2.3 Cast parts schema entities

### 5.2.3.1 Cast parts schema entity definitions

#### 5.2.3.1.1 activity\_execution\_result

An **activity\_execution\_result** is a type of **executed\_action** that defines the execution of a **process\_plan\_version** (see 4.2.238) to manufacture a cast part.

NOTE An **Activity\_execution\_result** is defined according to 4.2.2.

#### EXPRESS specification:

```
*)
ENTITY activity_execution_result
    SUBTYPE OF (executed_action);
    WHERE

    (* - to activity_result_record (as activity_results) -----*)

wr1: (NOT(SIZEOF(QUERY(ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
    (SIZEOF(QUERY(apr <* USEDIN(ap,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    ('CAST_PARTS_SCHEMA.ACTIVITY_RESULT_RECORD'
    IN TYPEOF(apr.representation))
    ))=1 ))) =0));

    (* ----- to sampled_set (as part_measured) -----*)

wr2: (SIZEOF(QUERY(ppa <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
    ((ppa.description='validation part shape') AND
    ('CAST_PARTS_SCHEMA.SAMPLED_SET'
    IN TYPEOF(ppa.property_or_shape))))=1);

    END_ENTITY;
    (*
```

#### Formal propositions:

**WR1:** There shall be exactly one reference by an **action\_property** through the **definition** attribute has exactly one reference through an **action\_property\_representation** through the **property** attribute that references an **activity\_result\_record** through the **representation** attribute.

**WR2:** There shall be exactly one reference by a **process\_property\_association** through the **process** attribute with a **description** attribute of 'validation part shape' and a **property\_or\_shape** attribute that references a **sampled\_set**.

### 5.2.3.1.2 activity\_result\_record

An **activity\_execution\_record** is a type of **representation** that defines a log of how a manufacturing operation was carried out.

NOTE An **Activity\_execution\_record** is defined according to 4.2.3.

#### EXPRESS specification:

```
*)
ENTITY activity_result_record
  SUBTYPE OF (representation);
  WHERE

  (* ----- date_and_time -----*)

  wr1: (SIZEOF(QUERY( dt <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS') |
    (dt\date_and_time_assignment.role.name='production date' )))=1);

  END_ENTITY;
  (*
```

#### Formal propositions:

**WR1:** There shall be exactly one reference by a **applied\_date\_and\_time\_assignment** through **items** that references **date\_and\_time\_role** with a **name** of 'production date' through **role**, and is of kind **date\_and\_time\_assignment** that references a **date\_and\_time** through **assigned\_date\_and\_time**.

### 5.2.3.1.3 allowed\_time

An **allowed\_time** is an **action\_property** that specifies an amount of time for the completion of an activity.

NOTE 1 An **Allowed\_time** is defined according to 4.2.5.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.1 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
  ENTITY allowed_time
    SUBTYPE OF (action_property);
WHERE

(* ----- allowance_factor----- *)

wr1: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF(QUERY ( it <* apr.representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF
  (it)) = 2) AND (it.name = 'allowance_factor')))=1)))=0));

(* ----- standard time----- *)

wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF(QUERY ( it <* apr.representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(
  it)) = 2) AND (it.name = 'standard time')) = 1) )) = 0);

(* ----- allowed_type----- *)

wr3: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF(QUERY ( it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'allowed_type'))<=1)))=0));

(* ----- allowed time source----- *)

wr4: SIZEOF(QUERY ( apr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF(QUERY ( it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'allowed time source')))=1)))=1;
END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the representation of an **allowed\_time** shall be of type **measure\_representation\_item** and **time\_measure\_with\_unit** with a **name** of 'allowance factor'.

**WR2:** There shall be zero or one **representation\_item** used for the representation of an **allowed\_time** shall be of type **measure\_representation\_item** and **time\_measure\_with\_unit** with a **name** of 'standard time'.

**WR3:** Exactly one **representation\_item** used for the representation of an **allowed\_time** shall be of type **descriptive\_representation\_item** with a **name** of 'allowed type'.

**WR4:** Exactly one **representation\_item** used for the representation of an **allowed\_time** shall be of type **descriptive\_representation\_item** with a **name** of 'allowed time source'.

### 5.2.3.1.4 alternate\_action\_method\_relationship

An **alternate\_action\_method\_relationship** is an **action\_method\_relationship** which relates an activity to an alternate activity.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.3 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
  ENTITY alternate_action_method_relationship
    SUBTYPE OF (ACTION_METHOD_RELATIONSHIP);
  WHERE
  wr1: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY'
    IN TYPEOF(SELF.related_method));
  wr2: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY'
    IN TYPEOF(SELF.relating_method));
  END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **related\_method** shall be of type **process\_plan\_activity**.

**WR2:** The **relating\_method** shall be of type **process\_plan\_activity**.

### 5.2.3.1.5 alternate\_plan\_relationship

An **alternate\_plan\_relationship** is an **action\_relationship** that relates a process plan with an alternate process plan.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.3 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY alternate_plan_relationship
  SUBTYPE OF (action_relationship);
  WHERE

  wr1: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
        IN TYPEOF(SELF.related_action));

  wr2: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
        IN TYPEOF(SELF.relying_action));
END_ENTITY;
(*)

```

Formal propositions:

**WR1:** The **relying\_action** and the **related\_action** shall be of type **process\_plan\_version**.

**WR2:** The **related\_action** shall be of type **process\_plan\_version** that is an alternate process plan.

**5.2.3.1.6 ancillary\_activity**

An **ancillary\_activity** is a type of **casting\_activity** which is not covered by any other category of activity.

NOTE 1 An **Ancillary\_activity** is defined according to 4.2.8.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.4 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY ancillary_activity
  SUBTYPE OF (process_plan_activity);
END_ENTITY;
(*)

```

**5.2.3.1.7 ancillary\_setup**

An **ancillary\_setup** is an **action\_method** which is not covered by any other category of setup activity.

NOTE 1 An **Ancillary\_setup** is defined according to 4.2.9.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.5 of ISO 10303-240:2005.

EXPRESS specification:

```
*)  
  ENTITY ancillary_setup  
    SUBTYPE OF (process_plan_activity);  
  END_ENTITY;  
(*
```

### 5.2.3.1.8 applied\_action\_assignment

An **applied\_action\_assignment** specifies those **action\_assignment\_items** to which a **action\_assignment** is assigned.

EXPRESS specification:

```
*)  
  ENTITY applied_action_assignment  
    SUBTYPE OF (action_assignment);  
    items: SET[1:?] OF action_item;  
  END_ENTITY;  
(*
```

Attribute definition:

**items:** the set of **action\_items** to which an **action\_assignment** is assigned.

### 5.2.3.1.9 applied\_action\_method\_assignment

An **applied\_action\_method\_assignment** specifies those **approved\_method\_items** to which an **action\_method** is assigned.

EXPRESS specification:

```
*)  
  ENTITY applied_action_method_assignment  
    SUBTYPE OF (action_method_assignment);  
    items: SET[1:?] OF action_method_item;  
  END_ENTITY;  
(*
```

Attribute definition:

**items:** the set of **action\_method\_items** to which an **action\_method\_assignment** is assigned.

### 5.2.3.1.10 applied\_action\_request\_assignment

An **applied\_action\_request\_assignment** specifies those **action\_request\_items** to which an **action\_request** is assigned.



EXPRESS specification:

```
*)
ENTITY applied_action_request_assignment
    SUBTYPE OF (action_request_assignment);
    items: SET[1:?] OF action_request_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **action\_request\_items** to which an **action\_request\_assignment** is assigned.

### 5.2.3.1.11 applied\_approval\_assignment

An **applied\_approval\_assignment** specifies those **approved\_items** to which an **approval** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_approval_assignment
    SUBTYPE OF (approval_assignment);
    items : SET [1:?] OF approved_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **approval\_items** to which an **approval\_assignment** is assigned.

### 5.2.3.1.12 applied\_classification\_assignment

An **applied\_classification\_assignment** specifies those **classification\_items** to which an **classification\_role** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_classification_assignment
    SUBTYPE OF (classification_assignment);
    items : SET [1:?] OF classification_item;
END_ENTITY; -- applied_classification_assignment
(*
```

Attribute definition:

**items:** the set of **classification\_items** to which an **classification\_assignment** is assigned.

### 5.2.3.1.13 applied\_date\_and\_time\_assignment

An **applied\_date\_and\_time\_assignment** specifies those **date\_and\_time\_items** to which an **date** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_date_and_time_assignment
    SUBTYPE OF (date_and_time_assignment);
    items: SET[1:?] OF date_and_time_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **date\_and\_time\_items** to which an **date\_and\_time\_assignment** is assigned.

### 5.2.3.1.14 applied\_date\_assignment

An **applied\_date\_assignment** specifies those **date\_items** to which an **date** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_date_assignment
    SUBTYPE OF (date_assignment);
    items: SET[1:?] OF date_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **date\_items** to which an **date\_assignment** is assigned.

### 5.2.3.1.15 applied\_document\_reference

An **applied\_document\_assignment** specifies those **document\_reference\_items** to which an **document** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_document_reference
    SUBTYPE OF (document_reference);
    items: SET[1:?] OF document_reference_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **document\_reference\_items** to which an **document** is assigned.

### 5.2.3.1.16 applied\_document\_usage\_constraint\_assignment

An **applied\_document\_usage\_constraint\_assignment** specifies those **document\_usage\_constraint\_item** to which an **document** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
  items : SET [1:?] OF document_reference_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **document\_usage\_constraint\_items** to which an **document\_usage\_constraint\_assignment** is assigned.

### 5.2.3.1.17 applied\_effectivity\_assignment

An **applied\_effectivity\_assignment** specifies those **effectivity\_items** to which an **effectivity** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_effectivity_assignment
  SUBTYPE OF (effectivity_assignment);
  items: SET[1:?] OF effectivity_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **effectivity\_items** to which an **effectivity\_assignment** is assigned.

### 5.2.3.1.18 applied\_external\_identification\_assignment

An **applied\_external\_identification\_assignment** specifies those **external\_identification\_items** to which an **external\_identification** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_external_identification_assignment
    SUPERTYPE OF (ONEOF (library_property_version_assignment,
        library_class_version_assignment))
    SUBTYPE OF (external_identification_assignment);
    items: SET[1:?] OF external_identification_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **external\_identification\_items** to which an **external\_identification\_assignment** is assigned.

### 5.2.3.1.19 applied\_group\_assignment

An **applied\_group\_assignment** specifies those **group\_items** to which an **group** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_group_assignment
    SUBTYPE OF (group_assignment);
    items : SET [1:?] OF group_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **group\_items** to which an **group** is assigned.

### 5.2.3.1.20 applied\_identification\_assignment

An **applied\_identification\_assignment** specifies those **identification\_assignment\_item** to which an **identification\_role** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_identification_assignment
    SUBTYPE OF (identification_assignment);
    items: SET[1:?] OF identification_assignment_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **identification\_assignment\_items** to which an **identification\_assignment** is assigned.

### 5.2.3.1.21 applied\_library\_assignment

An **applied\_library\_assignment** is a type of **applied\_classification\_assignment** that provides the means to reference information about a class within a parts library dictionary.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.11 of ISO 10303-240:2005.

#### EXPRESS specification:

```

*)
ENTITY applied_library_assignment
  SUBTYPE OF (applied_classification_assignment);
  WHERE

  (* library_part_assignment to class_BSU *)

  wr1: ((SELF\classification_assignment.role.name =
        'definitional class membership')) ;

  wr2: (((SELF\classification_assignment.assigned_class.name=
        'library identifier') AND
        ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_CLASS'
        IN TYPEOF (SELF\classification_assignment.assigned_class)))));

  (* library_part_assignment to property_value *)

  wr3: SIZEOF(QUERY ( edir <* USEDIN(SELF.assigned_class,
        'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATED_ITEM') |
        ((edir.name = 'name scope') AND
        ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_GENERAL_PROPERTY'
        IN TYPEOF(edir.relating_item))AND
        (SIZEOF(QUERY ( gpa <* USEDIN(edir.relating_item,
        'CAST_PARTS_SCHEMA.GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION') |
        ((gpa.name = 'definitional') AND
        (SIZEOF (QUERY (pdr <* USEDIN(gpa.derived_definition,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ((pdr.used_representation.name = 'property value')AND
        (SIZEOF(QUERY(ri <* pdr.used_representation.items|
        'CAST_PARTS_SCHEMA.VALUE_REPRESENTATION_ITEM'
        IN TYPEOF(ri))) =1 )))) = 1 )))) = 1))) = 1))) = 1;
END_ENTITY;
(*

```

Formal propositions:

**WR1:** The **role** attribute shall define a **classification\_role** with a **name** of 'definitional class membership'.

**WR2:** The **assigned\_class** attribute shall define a **group** of type **externally\_defined\_class** with a **name** of 'library identifier'

**WR3:** The **applied\_classification\_assignment** shall have definitional property pairs defined by one or more **value\_representation\_items** through a **representation** with **name** 'property value' defined through the **derived\_definition** for a **general\_property\_association** with a **name** of 'definitional' that is the **relating\_item** for a **externally\_defined\_item\_relationship** with a **name** of 'name scope'

### 5.2.3.1.22 applied\_organization\_assignment

An **applied\_organization\_assignment** specifies those **organization\_items** to which an **organization** is assigned.

EXPRESS specification:

```
*)  
ENTITY applied_organization_assignment  
    SUBTYPE OF (organization_assignment);  
    items: SET[1:?] OF organization_item ;  
END_ENTITY;  
(*
```

Attribute definition:

**items:** the set of **organization\_items** to which an **organization\_assignment** is assigned.

### 5.2.3.1.23 applied\_person\_and\_organization\_assignment

An **applied\_person\_and\_organization\_assignment** specifies those **person\_and\_organization\_items** to which a **person\_and\_organization** is assigned.

EXPRESS specification:

```
*)  
ENTITY applied_person_and_organization_assignment  
    SUBTYPE OF (person_and_organization_assignment);  
    items : SET [1:?] OF person_and_organization_item;  
END_ENTITY; -- applied_person_and_organization_assignment  
(*
```

Attribute definition:

**items:** the set of **person\_and\_organization\_items** to which an **person\_and\_organization\_assignment** is assigned.

### 5.2.3.1.24 applied\_security\_classification\_assignment

An **applied\_security\_classification\_assignment** specifies those **security\_classification\_items** to which an **security\_classification** is assigned.

EXPRESS specification:

```
*)
  ENTITY applied_security_classification_assignment
    SUBTYPE OF (security_classification_assignment);
    items : SET [1:?] OF security_classification_item;
  END_ENTITY;
  (*
```

Attribute definition:

**items:** the set of **security\_classification\_items** to which an **security\_classification\_assignment** is assigned.

### 5.2.3.1.25 applied\_time\_assignment

An **applied\_time\_assignment** specifies those **time\_items** to which an **time** is assigned.

EXPRESS specification:

```
*)
  ENTITY applied_time_assignment
    SUBTYPE OF (TIME_ASSIGNMENT);
    items : SET [1:?] OF time_item;
  END_ENTITY;
  (*
```

Attribute definition:

**items:** the set of **time\_items** to which an **time\_assignment** is assigned.

### 5.2.3.1.26 assembly

An **assembly** is a type of **manufacturing\_activity** that defines the assembly process with the part being cast.

NOTE An **Assembly** is defined according to 4.2.13.

EXPRESS specification:

```

*)
  ENTITY assembly
    SUBTYPE OF (non_casting_activity);
    WHERE

(* ----- pattern section assembly sequence ----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'pattern section assembly sequence')
  ))) >=0 )))>=1;

(* ----- foam pattern coating density ----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'foam pattern coating density')
  ))) <=1 )))>=1;
  END_ENTITY; -- assembly
(*

```

Formal propositions:

**WR31** Exactly one **representation\_item** used for the representation of an **allowed\_time** shall be of type **descriptive\_representation\_item** with a name of 'pattern section assembly sequence'.

**WR2:** Exactly one **representation\_item** used for the representation of an **allowed\_time** shall be of type **descriptive\_representation\_item** with a name of 'foam pattern coating density'.

### 5.2.3.1.27 bankout\_frame

A **bankout\_frame** is a type of **equipment** that defines the used to retain dry sand in which a fragile core is embedded for support.

NOTE A **Bankout\_frame** is defined according to 4.2.14.



EXPRESS specification:

```

*)
ENTITY bankout_frame
  SUBTYPE OF (core_equipment);
  WHERE
  (* material_type *)

  wr1:SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
      TYPEOF(pdr.representation)) AND
      (SIZEOF(QUERY (it <* pdr.representation.items |
        (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND ((it.name = 'material type')
            ))) =1 )))>=1;
END_ENTITY;
  (*

```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action\_property** through the **definition** attribute has exactly one reference through an **action\_property\_representation** through the **property** attribute that references a **representation** through the **representation** attribute, that references a **descriptive\_representation\_item** through the **items** attribute.

**5.2.3.1.28 bench\_inspection**

A **bench\_inspection** is a type of **inspection\_activity** that defines the activity of inspecting a cast part on a bench.

NOTE A **Bench\_inspection** is defined according to 4.2.16.

EXPRESS specification:

```

*)
  ENTITY bench_inspection
    SUBTYPE OF (inspection_activity);
  END_ENTITY;
  (*

```

**5.2.3.1.29 boundary\_condition**

A **boundary\_condition** is a type of **resource\_property** that defines the state of physical phenomena at the surface of a region.

NOTE A **Boundary\_condition** is defined according to 4.2.17.

EXPRESS specification:

```

*)
ENTITY boundary_condition
    SUBTYPE OF (action_method);
WHERE

    (* boundary_condition to numeric_parameter(as condition_value) *)
    (* boundary_condition to property_relationship(as condition_value) *)

wr1: (SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (NOT (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    ((SIZEOF(QUERY(pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY(mri <* pd.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP']
    * TYPEOF(mri)) =1 ) AND
    (mri.name='condition value' ))
    )) =1 )) )) =1 )) )) =1 )) )) =0 ));

(* boundary_condition to shape_aspect (as area_to) *)

wr2: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
    (SIZEOF(QUERY(eds <* ama.items |
    ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
    )) =1 )) ))=0));

END_ENTITY; -- boundary_condition
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the representation of an **boundary\_condition** shall be of type **measure\_representation\_item** or a **property\_relationship**. with a name of 'condition value'.

**WR2:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **shape\_aspect**.

### 5.2.3.1.30 cast\_part

A **cast\_part** is a type of **product\_definition** that defines the cast part being manufactured

NOTE A **Cast\_part** is defined according to 4.2.19.

EXPRESS specification:

```

*)
ENTITY cast_part
  SUBTYPE OF (product_definition);
  WHERE

  wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ( 'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE' IN TYPEOF (sa)
    )))>=1;

  END_ENTITY;
(*)

```

Formal propositions:

**WR1:** There shall be exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape**, and it shall be referenced by exactly one **shape\_aspect** through the **of\_shape** attribute that is of type **casting\_instanced\_feature**.

**5.2.3.1.31 cast\_part\_with\_rigging**

A **cast\_part\_with\_rigging** is a type of **product\_definition** that defines the cast part being manufactured with the attached rigging.

NOTE A **Cast\_part\_with\_rigging** is defined according to 4.2.20.

EXPRESS specification:

```

*)
ENTITY cast_part_with_rigging
  SUBTYPE OF (product_definition);
  WHERE

  wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ( 'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE' IN TYPEOF (sa)
    )))>=1;

  wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    (SIZEOF([
    'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE',
    'CAST_PARTS_SCHEMA.GATING_SYSTEM'] * TYPEOF(sa) = 2)))=1;

  END_ENTITY;
(*)

```

ISO 10303-223:2008 (E)

Formal propositions:

**WR1:** There shall be exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape**, and it shall be referenced by exactly one **shape\_aspect** through the **of\_shape** attribute that is of type **casting\_instanced\_feature**.

**WR2:** There shall be exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape**, and it shall be referenced by exactly one **shape\_aspect** through the **of\_shape** attribute that is of type **casting\_instanced\_feature** and **gating\_system**.

### 5.2.3.1.32 casting\_activity

The **casting\_activity** is a type of **manufacturing\_activity**, that is a type of **action\_method** which is the definition of a process planning **Activity**.

NOTE 1 A **Casting\_activity** is defined according to 4.2.21.

NOTE 2 This application interpreted model is harmonized with the **manufacturing\_activity** object from paragraph 5.2.3.1.63 of ISO 10303-240:2005.

EXPRESS specification:

\*)

```
ENTITY casting_activity
  SUBTYPE OF (manufacturing_activity);
  WHERE
```

```
(* ----- micro_plan_reference ----- *)
```

```
wr1: (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA' IN TYPEOF (eds))
  )) <=1 )) )<=1);
```

```
END_ENTITY;
```

```
(*
```

Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the **items** attribute **externally\_defined\_schema**.

### 5.2.3.1.33 casting\_design\_feature

The **casting\_design\_feature** is a type of **casting\_feature\_definition**, that defines design features on a cast part.

NOTE A **Casting\_design\_feature** is defined according to 4.2.22

#### EXPRESS specification:

```

*)
  ENTITY casting_design_feature
    SUBTYPE OF (casting_feature_definition);
    WHERE

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  ('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) )) = 1;

(* ***** casting_design_feature to orientation (as placement) ***** *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 ))))=1;

(* ----- draft ----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
  (it.name = 'draft')) )) =1 ))))=1;

(* ***** machine_stock ***** *)

wr4: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  ('CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF(co)) )) <= 1;

  END_ENTITY; -- casting_design_feature
(*)

```

Formal propositions:

**WR1:** The **casting\_feature\_definition** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** Exactly one **representation\_item** used for the implicit representation of a **feature\_definition** shall be of type **placement** with a **name** of 'orientation'. The **placement** shall define a reference location and orientation for the origin of the **feature\_definition**.

**WR3:** The implicit representation of the feature shall have at most one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'draft'.

**WR4:** There shall be exactly one reference from a **property\_definition** through the **definition** attribute of type **machine\_allowance**.

### 5.2.3.1.34 casting\_envelope\_requirement

A **casting\_envelope\_requirement** is a type of **customer\_casting\_requirement** that defines the is the minimum dimensional space required for the cast part to occupy.

NOTE A  **Casting\_envelope\_requirement** is defined according to 4.2.23.

EXPRESS specification:

\*)

```
ENTITY casting_envelope_requirement
  SUBTYPE OF (property_definition);
  WHERE
```

```
(* ----- to numeric_parameter(as part_height) -----*)
```

```
wr1:(NOT (SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF(pdr.used_representation)) AND
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
  (it.name = 'part height')) ) = 0 ) ) ) = 0));
```

```
(* ----- to numeric_parameter(as part_length) -----*)
```

```
wr2:(NOT (SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF(pdr.used_representation)) AND
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
  (it.name = 'part length')) ) = 0 ) ) ) = 0));
```

```
(* ----- to numeric_parameter(as part_width) -----*)

wr3: (NOT (SIZEOF(QUERY (pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) AND
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
(it.name = 'part width')) )) =0 )) )) =0));
END_ENTITY;
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'part height'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'part length'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'part width'.

### 5.2.3.1.35 casting\_equipment

An **casting\_equipment** is a type of **action\_resource** that defines equipment for process planning of cast parts activities.

NOTE 1 A **Casting\_equipment** is defined according to 4.2.110.

NOTE 2 This application interpreted model is harmonized with the entity machine from paragraph 5.2.3.1.58 of ISO 10303-240:2005.

### EXPRESS specification:

```
*)
ENTITY casting_equipment
  SUPERTYPE OF (ONEOF (core_equipment,ejector_equipment,shakeout,
    finishing_equipment,moulding_equipment,melting_equipment))
  SUBTYPE OF (machine);
WHERE

(* ----- equipment_name ----- *)

wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
(('CAST_PARTS_SCHEMA.ORGANIZATION'
IN TYPEOF(adr.assigned_organization)) AND
(adr.assigned_organization.description = 'equipment company name')
) )) = 1;
```

ISO 10303-223:2008 (E)

```
(* ----- equipment_type----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'equipment type')
  )))) =1 ))))>=1;

(* ----- equipment to controller (as controlled_by) ----- *)

wr3: SIZEOF(QUERY ( arr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE') |
  (('CAST_PARTS_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP' IN TYPEOF(arr)) AND
  ('CAST_PARTS_SCHEMA.CONTROLLER' IN TYPEOF(arr.relating_resource)
  ))) = 1;

(* ----- equipment to in_facility_location (as location) ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION' IN
    TYPEOF(pdr.representation)) AND
    (pdr.representation.name='location')))<=1;

  END_ENTITY; -- equipment
  (*
```

Formal propositions:

**WR1:** There shall exactly one references from a **applied\_organization\_assignment** through the **items** attribute that references an **organization** with a **description** of 'equipment company name' of through the **assigned\_organization**.

**WR2:** There shall be exactly one reference from a **resource\_property** through the **resource** attribute that is referenced by a **resource\_property\_representation** through the **property** attribute that references through the **representation** attribute a **representation** that references through the **items** attribute a **descriptive\_representation\_item** of name 'equipment type'.

**WR3:** There shall be one or more references from a **action\_resource\_relationship** of type **machine\_assembly\_relationship** through **related\_resource** that references a **controller** through **relating\_resource**.

**WR4:** There shall be exactly one reference from a **resource\_property** through the **resource** attribute that is referenced by a **resource\_property\_representation** through the **property** attribute that references through the **representation** attribute a **representation** of type **in\_facility\_location** with name 'location'.



### 5.2.3.1.36 casting\_feature\_definition

A **casting\_feature\_definition** is a type of **characterized\_object** that is a preconceived form pattern.

NOTE This application interpreted model is harmonized with the **feature\_definition** object from paragraph 4.2.17 of ISO 10303-522.

#### EXPRESS specification:

```

*)
ENTITY casting_feature_definition
    SUPERTYPE OF (ONEOF (sand_cast_design_feature,
                        die_design_feature,
                        ejector_design_feature,
                        gating_design_feature,
                        casting_design_feature,
                        investment_design_feature))
    SUBTYPE OF (characterized_object);
    WHERE

    WR1: SIZEOF (['CAST_PARTS_SCHEMA.EJECTOR_DESIGN_FEATURE',
                'CAST_PARTS_SCHEMA.CASTING_DESIGN_FEATURE',
                'CAST_PARTS_SCHEMA.GATING_DESIGN_FEATURE',
                'CAST_PARTS_SCHEMA.DIE_DESIGN_FEATURE',
                'CAST_PARTS_SCHEMA.INVESTMENT_DESIGN_FEATURE',
                'CAST_PARTS_SCHEMA.SAND_CAST_DESIGN_FEATURE'])
        * TYPEOF (SELF) <= 1;

END_ENTITY;
(*)

```

#### Formal propositions:

**WR1:** The **feature\_definition** may be either a **ejector\_design\_feature**, **casting\_design\_feature**, **gating\_design\_feature**, **die\_design\_feature**, **investment\_design\_feature** or **sand\_cast\_design\_feature**.

### 5.2.3.1.37 casting\_equipment\_process

The **casting\_equipment\_process** is a type of **machining\_process** that defines the type of equipment, equipment setup, and activity required to perform a automated casting process.

NOTE 1 A **Equipment\_proces** is defined according to 4.2.111 and defines the requirement for **casting\_equipment\_proces**.

NOTE 2 This application interpreted model is harmonized with the entity **machining\_process** from paragraph 5.2.3.1.86 of ISO 10303-240:2005.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
  ENTITY casting_equipment_process
    SUBTYPE OF (machining_process);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
      'CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(sar)))=1;
  END_ENTITY;
  (*
```

Formal propositions:

**WR1:** There shall be one or more references through the **usage** attribute that references **casting\_equipment**.

### 5.2.3.1.38 casting\_equipment\_setup

An **casting\_equipment\_setup** is an **process\_plan\_activity** which places or locates the cast part on the equipment.

NOTE 1 A **Equipment\_setup** is defined according to 4.2.113 and defines the requirement for **casting\_equipment\_setup**.

NOTE 2 This application interpreted model is harmonized with the entity **machine\_setup** from paragraph 5.2.3.1.60 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
  ENTITY casting_equipment_setup
    SUBTYPE OF (machine_setup);
  WHERE

    wr1: (NOT(SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
      (('CAST_PARTS_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(sar)) AND
      (SIZEOF(QUERY( equ <* sar.resources |
      ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(equ))))
      =1 ))))=0));

  END_ENTITY;
  (*
```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action\_resource\_requirement** of type **requirement\_for\_action\_resource** through the **operations** that references one or more **casting\_equipment** through **resources**.

### 5.2.3.1.39 casting\_feature\_size

A **casting\_feature\_size** is a type of **casting\_feature\_definition** that defines the implicit attributes of the shape of a casting feature.

NOTE An **Item\_size** is defined according to 4.2.163 and defines the requirements for **casting\_feature\_size**.

#### EXPRESS specification:

```

*)
ENTITY casting_feature_size
  SUBTYPE OF (shape_aspect);
WHERE

(* ----- item_size to numeric_parameter (as item_height)----- *)

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'item height')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_length)----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'item length')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_width)----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'item width')) ) <=1 )))=1;

```

ISO 10303-223:2008 (E)

```
(* ----- item_size to numeric_parameter (as item_diameter)----- *)  
  
wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |  
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN  
    TYPEOF(pdr.used_representation)) AND  
    (SIZEOF(QUERY (it <* pdr.used_representation.items |  
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',  
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']  
        * TYPEOF(it)) = 2) AND  
        (it.name = 'item diameter')) ) ) <=1 )))=1;  
  
(* ----- item_size to numeric_parameter (as item_draft)----- *)  
  
wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |  
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN  
    TYPEOF(pdr.used_representation)) AND  
    (SIZEOF(QUERY (it <* pdr.used_representation.items |  
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',  
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']  
        * TYPEOF(it)) = 2) AND  
        (it.name = 'item draft')) ) ) <=1 )))=1;  
  
END_ENTITY;  
(*
```

Formal propositions:

**WR1:** At most one **representation\_item** used for the implicit representation of a **casting\_feature\_size** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'height'.

**WR2:** At most one **representation\_item** used for the implicit representation of a **casting\_feature\_size** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'length'.

**WR3:** At most one **representation\_item** used for the implicit representation of a **casting\_feature\_size** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'width'.

**WR4:** At most one **representation\_item** used for the implicit representation of a **casting\_feature\_size** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'diameter'.

**WR5:** At most one **representation\_item** used for the implicit representation of a **casting\_feature\_size** shall be of type **measure\_representation\_item** and **plane\_angle\_measure\_with\_unit** with a **name** of 'draft'.

### 5.2.3.1.40 casting\_insert

A **casting\_insert** is a type of **casting\_product\_definition** that is the secondary material, usually a metal, that are placed in a mould and cast into the body of the casting and appears as integral structural parts of the final casting.

NOTE A **Casting\_insert** is defined according to 4.2.24.

#### EXPRESS specification:

```

*)
ENTITY casting_insert
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ***** insert_name ***** *)

  wr1 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'insert name')))) =1 )))=1;

(* ***** material_type ***** *)

  wr2: (SIZEOF(QUERY (co <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

(* ***** positioning_method ***** *)

  wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'positioning method')))) =1 )))=1;

END_ENTITY;
(*

```

Formal propositions:

WR1: Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'insert name'.

WR2: There shall be exactly one reference from a **material\_designation** through the **definitions** attribute.

WR3: Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'positioning method'.

### 5.2.3.1.41 casting\_instanced\_feature

A **casting\_instanced\_feature** is a type of **shape\_aspect** and of **casting\_feature\_definition** that is the representation of a preconceived pattern for the casting application purpose.

NOTE This application interpreted model is harmonized with the feature\_definition object from paragraph 4.2.23 of ISO 10303-522.

EXPRESS specification:

```
*)
ENTITY casting_instanced_feature
  SUBTYPE OF (casting_feature_definition, shape_aspect);
WHERE
  WR1: 'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN
        TYPEOF (SELF.of_shape.definition);
  WR2: SELF.product_definitional;
END_ENTITY;
(*
```

Formal propositions:

WR1: The **instanced\_feature** shall be an aspect of the shape of a **product\_definition**.

WR2: The **instanced\_feature** shall lie on the boundary of the part.

Informal propositions:

IP1: If an **casting\_instanced\_feature** has a **shape\_representation** and if the **product\_definition\_shape** that is referenced as **of\_shape** by the **casting\_instanced\_feature** has a **shape\_representation** then these **shape\_representations** shall have the same **geometric\_representation\_context**.

### 5.2.3.1.42 casting\_method

A **casting\_method** is a type of **action\_method** that defines the moulding process to be used for the cast part.

EXPRESS specification:

```

*)
  ENTITY casting_method
    SUBTYPE OF (action_method);
  END_ENTITY;
(*

```

**5.2.3.1.43 casting\_per\_order**

A **casting\_per\_order** is a type of **casting\_verification** that is the required number of production castings per order to be evaluated for compliance to the design record and required specifications.

NOTE A **Casting\_per\_order** is defined according to 4.2.25.

EXPRESS specification:

```

*)
  ENTITY casting_per_order
    SUBTYPE OF (casting_verification);
  WHERE

  wr1: (NOT(SIZEOF(QUERY( ad <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_DIRECTIVE.REQUESTS') |
    (NOT(SIZEOF(QUERY( da <* USEDIN(ad,
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION.DIRECTIVE') |
    (('CAST_PARTS_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(da)) AND
    (SIZEOF(QUERY(ppa <* USEDIN(da,
    'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
    ('CAST_PARTS_SCHEMA.SAMPLED_SET') IN TYPEOF(ppa.property_or_shape))
    =1)) )) =0 )) =0 ));

  END_ENTITY;

(*

```

Formal propositions:

WR1: There shall be exactly one reference from an **action\_directive** through the **requests** attribute that is referenced by exactly one reference from a **directed\_action** through the **directive** attribute that is of type **property\_process** that is referenced from a **property\_association** through the **process** attribute that references a **sampled\_set** through the **property\_or\_shape** attribute.

**5.2.3.1.44 casting\_process**

A **casting\_process** is a type of **casting\_activity** that defines the processing activity to interact with the material part being casted.

NOTE A **Casting\_process** is defined according to 4.2.26.

EXPRESS specification:

```

*)
ENTITY casting_process
    SUBTYPE OF (casting_activity);
WHERE

(* ----- core_making_process ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'core making process') AND
    (it.description IN ['shell','no bake','lost form','gas cured',
    'green sand'])))) =1 ))))>=1;

wr2: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'core making process') AND
    (it.description IN ['core presentation', 'core setting',
    'die assembly', 'die handling',
    'die temperature control', 'furnace loading',
    'gate system removal', 'grinding',
    'inspection', 'ladle loading',
    'loading injection system', 'melt raw material',
    'mould closing', 'mould making process',
    'package and shipping', 'part routing',
    'pattern handling', 'tool handling',
    'pouring process', 'refractor removal',
    'removal from die', 'shakeout',
    'slurry process', 'sand removal',
    'wax removal', 'shell',
    'green sand', 'no bake',
    'lost form', 'vacuum',
    'static', 'tilt',
    'low pressure'])))) =1 ))))>=1;

wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    (('CAST_PARTS_SCHEMA.DEFINING_ACTION_METHOD_RELATIONSHIP'
    IN TYPEOF(ap)) AND
    ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT_SETUP'
    IN TYPEOF(ap.related_method))))=1;

```



```

(* ----- casting_process to mismatch_tolerance (as tolerances)---- *)

wr4: (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.MISMATCH_TOLERANCE' IN TYPEOF (eds))
  )) <=1 )) )<=1);

(* --- casting_process to production_core_box (as core_to_be_made)---- *)

wr5:(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX' IN TYPEOF (eds.definition))
  )) =1 )) ) =0);

(* ---- casting_process to numeric_parameter (as process_parameter) ----- *)

wr6: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP'] * TYPEOF
  (it)) = 1) AND (it.name = 'special instruction'))
  )) =1 ))) >=1;

(* ---- casting_process to shape_aspect (as described_by) ----- *)

wr7: (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
  )) <=1 )) )>=0);

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core making process' and a **description** of either 'shell', 'no bake', 'lost form', 'gas cured' or 'green sand'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'type of process' and a **description** of either 'core presentation', 'core setting', 'die assembly', 'die handling', 'die temperature control', 'furnace loading', 'gate system removal', 'grinding', 'inspection', 'ladle loading', 'loading injection system', 'melt raw material', 'mould closing', 'mould making process', 'package and shipping', 'part handling', 'pattern handling', 'pouring process', 'refractor removal', 'removal from die', 'shakeout', 'slurry process', 'sand removal', 'wax removal', 'shell', 'green sand', 'no bake', 'lost form', 'vacuum', 'static' or 'tilt', 'low pressure'.

**WR3:** Exactly one reference that is the **related\_method** in exactly one **action\_method\_relationship** in which the **relating\_method** is an **casting\_equipment\_setup**.

**WR4:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment**, the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is exactly one **mismatch\_tolerance**.

**WR5:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment**, the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is exactly one **property\_definition** that references a **production\_core\_box** through the **definition** attribute.

**WR6:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** of a **property\_relationship** with a **name** of 'special instruction'.

**WR7:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment**, the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is exactly one **shape\_aspect**.

### 5.2.3.1.45 casting\_product\_definition

A **casting\_product\_definition** is a type of **product\_definition** that defines the a casting product definition.

EXPRESS specification:

\*)

```
ENTITY casting_product_definition
    SUBTYPE OF (product_definition);
    WHERE
    WR1: 'CAST_PARTS_SCHEMA.APPLICATION_CONTEXT_ELEMENT'
        IN TYPEOF(SELF.frame_of_reference);
    WR2: SELF.frame_of_reference.name = 'function definition';
```

```
(* --- tooliong to cast_part (as tooling designed_for)---- *)

wr3: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CAST_PART'
  IN TYPEOF(pdr.relatng_product_definition)))=1;

(* --- tooliong to date ( as date_last_used)---- *)

wr4: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (ada.role.name='date last used')))=1);
  END_ENTITY; -- casting_product_definition
(*
```

#### Formal propositions:

**WR1:** Exactly one reference that is the **frame\_of\_reference** in exactly one **application\_context\_element**.

**WR2:** Exactly one reference that is the **frame\_of\_reference** in exactly one **application\_context\_element** with **name** of 'function definition'.

**WR3:** There shall be exactly one reference from an **product\_definition\_relationship** through the **related\_product\_definition** attribute that references a **cast\_part** through the **relating\_product\_definition**.

**WR4:** There shall be exactly one reference from an **applied\_date\_assignment** through the **items** attribute that is of type **date\_assignment** that references through the **role** attribute a **date\_role** with **name** of 'date last used'.

#### **5.2.3.1.46 casting\_service\_data**

A **casting\_service\_data** is a type of **customer\_casting\_requirement** that is the information that describes the requirements and types of usage application of the cast part in service.

NOTE A  **Casting\_service\_data** is defined according to 4.2.27.

EXPRESS specification:

```

*)
ENTITY casting_service_data
    SUBTYPE OF (customer_casting_requirement);
WHERE

(* ----- to corrosion_service (as corrosion_service_requirement)----- *)

wr1:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.CORROSION_SERVICE'
    IN TYPEOF(pdr.related_property_definition))
    ))>=0);

(* ----- to descriptive_parameter(as comments) ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'comments')
    ) ) =1 )))=0));

(* ----- wear_with_lubrication ----- *)

wr3:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'wear or abrasion lubrication') AND
    (it.description IN ['good','intermitten','none'] )
    ) )=1 ))) = 0);

(* -- to elevated_temperature_service (as elevated_temperature_service) *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.ELEVATED_TEMPERATURE_SERVICE'
    IN TYPEOF(pdr.related_property_definition))
    ))=1);

(* finishing_or_machining_operations(machining_and_finishing_requirements) *)

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.FINISHING_AND_MACHINING_OPERATION'
    IN TYPEOF(pdr.related_property_definition))
    ))=1);

```

```
(* ----- to material (as subject_to_abrasive_wear_by)----- *)

wr6:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'subject to abrasize wear against')
  ) ) =1 )))=0));

(* ----- to material (as subject_to_wear_against)----- *)

wr7:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'subject to wear against')
  ) ) =1 )))=0));

(* ---to mechanical_stress (as mechanical_stress_requirements) ----- *)

wr8:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.MECHANICAL_STRESS'
  IN TYPEOF(pdr.related_property_definition)
  ))=1);

END_ENTITY;
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **relating\_action\_property** in exactly one **action\_property\_-relationship** in which the **related\_action\_property** is a **corrosion\_service**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'comments'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'wear or abrasion lubrication' and a **description** of 'good','intermitten','none'.

**WR4:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_-definition\_relationship** in which the **related\_property\_definition** is a **elevated\_temperature\_service**.

**WR5:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_-definition\_relationship** in which the **related\_property\_definition** is a **finishing\_and\_machining\_operation**.

ISO 10303-223:2008 (E)

**WR6:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'subject to abrasize wear against'.

**WR7:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'subject to wear against'.

**WR8:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** in which the **related\_property\_definition** is a **mechanical\_stress**.

### 5.2.3.1.47 casting\_verification

A **casting\_verification** is a type of **versioned\_action\_request** that is the documented proof that a cast part has satisfied or failed to satisfy all design record requirements and specifications for the part as designed.

NOTE A **Casting\_verification** is defined according to 4.2.28.

#### EXPRESS specification:

\*)

```
ENTITY casting_verification
  SUBTYPE OF (versioned_action_request);
  WHERE
```

```
(* ----- cavity_impression_id -----*)
```

```
wr1: (NOT (SIZEOF (QUERY (ars <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
  (NOT (SIZEOF (QUERY (ap <* USEDIN (ars.method,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (it)) AND
  (it.name = 'cavity impression id'))
  )) =1 ))) =0 ))) =0 ))) =0 ))) =0 ));
```

```
(* --- casting_verification to tolerance_requirement (as verify_with) --- *)
```

```
wr2: SIZEOF (QUERY (ada <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF (ada)) AND
  (NOT (SIZEOF (QUERY (ari <* ada.items |
  'CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
  IN TYPEOF (ari)) =1)))) =0;
```

```
(* ----- to quality_acceptance_report (as destruction) ----- *)

wr3: SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT(SIZEOF(QUERY(ari <* ada.items |
    (('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF(ari)) AND
    (ari.kind.product_data_type='quality acceptance report') AND
    (ari\document.description='destruction'))
    )) =1))))=0;

(* ----- to quality_acceptance_report (as nde) ----- *)

wr4: SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT(SIZEOF(QUERY(ari <* ada.items |
    (('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF(ari)) AND
    (ari.kind.product_data_type='quality acceptance report') AND
    (ari\document.description='nde'))
    )) =1))))=0;
  END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** in which the **representation** is a **representation** in which exactly one **items** is a **descriptive\_representation\_item** with a **name** of 'cavity impression id'.

**WR2:** Exactly one reference that is the **assigned\_action\_request** in exactly one **action\_request\_assignment** is of type **applied\_action\_request\_assignment** in which the **items** is exactly one **tolerance\_requirement**

**WR3:** Exactly one reference that is the **assigned\_action\_request** in exactly one **action\_request\_assignment** is of type **applied\_action\_request\_assignment** in which the **items** is exactly one **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'quality acceptance report', and the **document** attribute of **description** is 'destruction'.

**WR4:** Exactly one reference that is the **assigned\_action\_request** in exactly one **action\_request\_assignment** is of type **applied\_action\_request\_assignment** in which the **items** is exactly one **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'quality acceptance report', and the **document** attribute of **description** is 'nde'.

### 5.2.3.1.48 chaplet

A **chaplet** is a type of **casting\_product\_definition** that defines a metal support that stabilizes the core in the mould when the core seat or the core strength is inadequate to support the core.

NOTE A **Chaplet** is defined according to 4.2.30.

#### EXPRESS specification:

```

*)
ENTITY chaplet
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ***** chaplet_type ***** *)

wr1:  SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'chaplet type')))) =1 )))=1;

(* ----- chaplet to item_size (as chaplet_size) ----- *)

wr2:  SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='chaplet reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chaplet dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 ))) >=1;

(* ***** chaplet to chaplet_pad (as pad) ***** *)

wr3:  SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='chaplet reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chaplet pad reference usage') AND
  ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 ))) >=1;
END_ENTITY;
(*

```



Formal propositions:

**WR1:** Exactly one **representation\_item** used for the representation shall be of type **descriptive\_representation\_item** with a **name** of 'chaplet type'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'chaplet dimension occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'chaplet dimension usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'chaplet pad reference occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chaplet pad reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chaplet\_pad**.

### 5.2.3.1.49 checking\_aid\_tool

A **checking\_aid\_tool** is a type of **casting\_method** that defines the device or an implement used during the inspection process that provides assistance in the checking procedures.

NOTE A **Checking\_aid\_tool** is defined according to 4.2.32.

EXPRESS specification:

\*)

```
ENTITY checking_aid_tool
  SUBTYPE OF (customer_casting_requirement);
WHERE
```

```
(* ----- aid id ----- *)
```

```
wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'aid id')
  ) ))=1 )))) = 0);
```

```
(* ----- aid name----- *)
```

```
wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'aid name')
  ) ))=1 )))) = 0);
```

ISO 10303-223:2008 (E)

```
(* ----- aid_tool ----- *)

wr3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'aid tool') AND
  (it.description IN ['fixture','guage','paper document','template'] )
  ) )=1 )))) = 0);

(* ----- revision id ----- *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'revision id')
  ) )=1 )))) = 0);

(* --- checking_aid_tool to date ( as revision_date)----- *)

wr5: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (ada.role.name='revision date'))=1);

  END_ENTITY; -- checking_aid_tool
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the representation shall be of type **descriptive\_representation\_item** with a **name** of 'aid id'.

**WR2:** Exactly one **representation\_item** used for the representation shall be of type **descriptive\_representation\_item** with a **name** of 'aid name'.

**WR3:** Exactly one **representation\_item** used for the representation shall be of type **descriptive\_representation\_item** with a **name** of 'aid tool' and **description** of 'fixture','guage','paper document' or 'template'.

**WR4:** Exactly one **representation\_item** used for the representation shall be of type **descriptive\_representation\_item** with a **name** of 'revision id'.

**WR5:** Exactly one reference that is the **items** in exactly one **applied\_date\_assignment** that references through the **role** attribute **date\_role** with **name** of 'revision date'.

### 5.2.3.1.50 chaplet\_pad

A **chaplet\_pad** is a type of **casting\_product\_definition** that is a shallow projection attached to the cope and drag surfaces of a pattern and core box that are opposite of the parting surfaces.

NOTE A **Chaplet\_pad** is defined according to 4.2.31.

#### EXPRESS specification:

```
*)
ENTITY chaplet_pad
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

  wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='chaplet pad dimension occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'chaplet pad dimension usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relatng_shape_aspect))
    )) =1 )))) =1;
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'chaplet pad dimension occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'chaplet pad dimension usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.51 chill

A **chill** is a type of **casting\_product\_definition** that is a metallic insert placed in the mould to promote rapid cooling at a desired point to make the casting harder at this point or to promote directional solidification and prevent shrinkage at a hot spot which cannot be fed from a riser.

NOTE A **Chill** is defined according to 4.2.33.

EXPRESS specification:

```

*)
ENTITY chill
  SUBTYPE OF (sand_cast_design_feature);
WHERE

(* ----- internal_or_external_chill----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'internal or external chill') AND
    ((it.description = 'true') OR (it.description = 'false')
    )))) =1 )))) >=1;

(* ----- material_type ----- *)

wr2: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  ('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) )) =1;

(* ----- chill to shape_element (as chill_area_shape)----- *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='chill area occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chill area shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) =1;

END_ENTITY; -- chill
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'internal or external chill' and a **description** of either 'true' or 'false'.

**WR2:** Exactly one reference that is the reference that is the **definition** in exactly one **material\_designation**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'chill area occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chill area shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.52 choke

A **choke** is a type of **casting\_product\_definition** that is the part of the gating system that most restricts or regulates the flow of metal into the mould cavity.

NOTE A **Choke** is defined according to 4.2.34.

#### EXPRESS specification:

```

*)
    ENTITY choke
        SUBTYPE OF (gating_design_feature);
        WHERE

(* ----- choke to item_size (as core_vent_size) ----- *)

WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
    ((sa.description='choke occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'choke dimension usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) =1 )))) >=1;

(* ----- choke to location_element (as vent_placement)----- *)

wr2 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
    (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
    TYPEOF(pdr.used_representation)) AND
    (pdr.used_representation.name='location'))))=1;
END_ENTITY; -- choke
(*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'choke occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'choke reference usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **location\_shape\_representation** through the **used\_representation** that has a **name** of 'location'.

### 5.2.3.1.53 class

A **class** is a type of **group** that specifies a type of classification assignment.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.24 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
ENTITY class
  SUBTYPE OF (group);
END_ENTITY;
(*
```

### 5.2.3.1.54 composition\_element

A **compositon\_element** is a type of **material\_property** that defines the numer of elements in a material.

NOTE A **Composition\_element** is defined according to 4.2.37.

EXPRESS specification:

```
*)
ENTITY composition_element
  SUBTYPE OF (material_property);
  WHERE
    (* --- element----- *)
```

```
WR1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(dri <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(dri)) AND
  (dri.name = 'element'))
  ))=1)))=0));
```

```
(* ---  compositon_element to numeric_range (as amount)----- *)

WR2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(dri <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT',
'CAST_PARTS_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
* TYPEOF(dri)) = 3) AND (dri.name = 'amount')) ) =1)))=0));

END_ENTITY; -- composition_element
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'element'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item**, **length\_measure\_with\_unit** and **qualified\_representation\_item** with a **name** of 'amount'.

### 5.2.3.1.55 composition\_element\_record

A **composition\_element\_record** is a type of **activity\_execution\_record** that is a record of the quantitative measure of a chemical element in a material.

NOTE A **Composition\_element\_record** is defined according to 4.2.38.

### EXPRESS specification:

```
*)
ENTITY composition_element_record
  SUBTYPE OF (activity_result_record);
  WHERE

  (* -- to numeric_parameter (as actual_composition_amount) -----*)

wr1: (SIZEOF(QUERY( it <* SELF.items |
(SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
(it.name = 'actual_composition amount'))=1);
```

ISO 10303-223:2008 (E)

```
(* -----to compositon_element (as recorded_usage) -----*)  
  
wr2: (SIZEOF(QUERY( pdr <* USEDIN(SELF,  
    'CAST_PARTS_SCHEMA.'+  
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |  
    ('CAST_PARTS_SCHEMA.COMPOSITION_ELEMENT'  
    IN TYPEOF(pdr.definition))))=1);  
  
    END_ENTITY;  
    (*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_ - representation\_item** and **length\_measure\_with\_unit** with a **name** of 'amount'.

**WR2:** There shall be exactly one reference through the **used\_representaion** attribute by **property\_ - definiton\_representation** that references a **property\_definition** of type **composition\_element** through the **definition** attribute.

### 5.2.3.1.56 composition\_inspection

A **compositon\_inspection** is a type of **inspection\_or\_test\_result** that defines the outcome of a measurement of the requested **composition\_requirement** with the actual material composition.

NOTE A **Composition\_inspection** is defined according to 4.2.39.

EXPRESS specification:

```
*)  
    ENTITY composition_inspection  
        SUBTYPE OF (inspection_or_test_result);  
        WHERE  
  
        (* ----- to material (as composition_result) -----*)  
  
wr1: ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (SELF.definition));  
  
    (* -----to material_property (as requirements)----- *)  
  
wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,  
    'CAST_PARTS_SCHEMA.'+  
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |  
    ('CAST_PARTS_SCHEMA.MATERIAL_PROPERTY'  
    IN TYPEOF(pdr.related_property_definition)  
    ))=1);  
  
    END_ENTITY;  
    (*
```



Formal propositions:

**WR1:** There shall be exactly one reference through the **definition** attribute to a **product\_definition**.

**WR2:** There shall be one or more references through the **relating\_property\_definition** by a **property\_definition\_relationship** that references through the **related\_property\_definition** a **property\_definition** of type **material\_property**.

### 5.2.3.1.57 condition\_or\_assumption

A **condition\_or\_assumption** is a type of **resource\_property** that is a state of physical phenomena or presumption about the state of physical phenomena.

NOTE A **Condition\_or\_assumption** is defined according to 4.2.43.

EXPRESS specification:

\*)

```
ENTITY condition_or_assumption
  SUBTYPE OF (action_method);
WHERE
```

```
(* condition_or_assumption to numeric_parameter(as associated_property) *)
wr1:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri))))=1)
  ))=1))))=0))))=0));
```

```

(* to property_relationship(as associated_property) *)

wr2: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT (SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN TYPEOF(mri))))=1)
  ))=1))))=0))))=0));

(* condition_or_assumption to CAST_PARTS_SCHEMA (as further_information) *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
  (adr.assigned_document))))=1;

END_ENTITY; -- condition_or_assumption
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item**.

**WR2:** There shall be exactly one reference through the **property** attribute to a **property\_relationship** that references a **property\_relationship** through the **representation** attribute.

**WR3:** There shall be exactly one reference through the **items** attribute to a **applied\_document\_reference** that references a **document** through the **assigned\_document** attribute.

**5.2.3.1.58 continuous\_process\_relationship**

The **continuous\_process\_relationship** is a type of **sequential\_method** that relates a **casting\_process** to a **casting\_process**.

NOTE 1 A **Continuous\_process** is defined according to 4.2.45 and defines the requirement for **continuous\_process\_relationship**.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.25 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY continuous_process_relationship
    SUBTYPE OF (sequential_method);

WHERE

wr1: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
    IN TYPEOF(SELF.related_method));

wr2: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
    IN TYPEOF(SELF.relying_method));

wr3: (SELF.description IN ['serial','batch', 'serial and batch']);

END_ENTITY;
(*)

```

Formal propositions:

**WR1:** The **related\_method** attribute shall be **casting\_process**.

**WR2:** The **relying\_method** attribute shall be **casting\_process**.

**WR3:** The **description** shall be one of 'serial','batch', or 'serial and batch'.

**5.2.3.1.59 controller**

A **controller** is a type of **action\_resource** that is the computerized numerical controller that supplies input to **equipment**.

NOTE 1 A **Controller** is defined according to 4.2.46.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.26 of ISO 10303-240:2005.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY controller
  SUBTYPE OF (action_resource);
WHERE

(* -- controller to specification (as controller_specification) *)

wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN TYPEOF
  (adr.assigned_document))))=1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **controller** shall be in the set of **items** of exactly one **applied\_document\_reference** with a **assigned\_document** defines the **process\_plan\_specification** containing the controller specification.

### 5.2.3.1.60 cooling\_port

A **colling\_port** is a type of **casting\_feature\_definition** that is an inlet and outlet that permits a liquid or air to pass through a die mould to quickly reduce the mould temperature.

NOTE A **Cooling\_port** is defined according to 4.2.47.

EXPRESS specification:

```
*)
ENTITY cooling_port
  SUBTYPE OF (die_design_feature);
WHERE

(* ----- male_or_female----- *)

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'male or female') AND
  ((it.description IN ['male','female'] ) ) ) ) ) ) ) ) )>=1; )>=1;
```

```
(* ----- cooling_port to item_size (as cooling_size) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='cooling port occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'cooling port size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1;

(* ----- cooling_port to planar_element (as port_location)----- *)

wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='port location')))=1;
END_ENTITY; -- cooling_port
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'male or female' and a **description** of either 'male' or 'female'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'cooling port occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'cooling port size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **planar\_shape\_representation** through the **used\_representation** that has a **name** of 'port location'.

#### 5.2.3.1.61 core

A **core** is a type of **casting\_feature\_definition** that is a design of specially formed material inserted into a mould to shape the interior or other part of a casting that cannot be easily shaped by the pattern.

NOTE A **Core** is defined according to 4.2.48.

EXPRESS specification:

```

*)
ENTITY core
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- core_type----- *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'core type') AND
  ((it.description IN ['baked core','cover core','isocure','ram up core',
  'ring core','setup core','slab core','so2 core'] )
  )))) =1 ))))>=1;

(* ----- material_definition ----- *)

WR2: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)) AND
  (co.name IN ['ceramic','oil bonded sand','plaster','resin coated sand']
  )) ) ) =1);

(* ----- core to location_element (as vent_placement)----- *)

wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='vent location')))=1;

(* ----- core to sand_mould (as core_placement) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core placement reference usage') AND
  ('CAST_PARTS_SCHEMA.SAND_MOULD' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- core to vent ( as gas vent) ----- *)

WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gas vent reference usage') AND
  ('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

END_ENTITY; -- core
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core type' and **description** of 'ram up core','set core'.

**WR2:** Exactly one reference that is the reference that is the **definition** in exactly one **material\_designation**.

**WR3:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **location\_shape\_representation** through the **used\_representation** that has a **name** of 'vent location'.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'core occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core placement reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **sand\_mould**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'core occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'gas vent reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **vent**.

### 5.2.3.1.62 core\_assembly

A **core\_assembly** is a type of **casting\_method** that is when each core for the casting is placed in the proper core print set.

NOTE A **Core\_assembly** is defined according to 4.2.49.

EXPRESS specification:

```

*)
ENTITY core_assembly
  SUBTYPE OF (casting_method);
  WHERE

  (* ----- core_assembly to descriptive_parameter (as hard_form) -----*)

  wr1: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'hard form')
    ))) =1 ))))>=1;

  (* ----- core_assembly to proof_report (as soft_copy) -----*)

  wr2: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF(SELF)) AND (NOT (SIZEOF(QUERY ( doc <* SELF.
    documents | (doc.kind.product_data_type = 'proof report') ))
    = 0));

  END_ENTITY;
  (*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'hard form'.

**WR2:** **Core\_assembly** shall be of type **action\_method\_with\_associated\_documents** and shall be zero or more references to a **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'proof report'.

### 5.2.3.1.63 core\_bench

A **core\_bench** is a type of **equipment** that is a place where small cores are made, often referred to as bench cores.

NOTE A **Core\_bench** is defined according to 4.2.50.



EXPRESS specification:

```

*)
ENTITY core_bench
    SUBTYPE OF (core_equipment);

WHERE
(* ----- core_required----- *)
wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'core required')) ) =1 )))=1;

(* ----- wax_vent----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'wax vent'
    )))) =1 ))))>=1;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_ - representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core required'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'wax vent'.

**5.2.3.1.64 core\_box\_tooling**

A **core\_box\_tooling** is a type of **casting\_product\_definition** that defines the tooling for a **production\_ - core\_box**.

NOTE A **Core\_box\_tooling** is defined according to 4.2.51.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY core_box_tooling

    SUBTYPE OF (casting_product_definition);
    WHERE

(* core_box_tooling to production_core_box(as cores_for_pattern)  *)

wr1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='core box tooling occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'production core box reference usage') AND
    ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
    IN TYPEOF(sar.relatng_shape_aspect))
    )) >=1 )))) >=1;

END_ENTITY; -- core_box_tooling
(*
```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'core box tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core box reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_core\_box**.

### 5.2.3.1.65 core\_box\_vent

A **core\_box\_vent** is a type of **representation** that is a perforated object that is inserted into a core box to permit air to escape as core sand it forced into the core box under pressure.

NOTE A **Core\_box\_vent** is defined according to 4.2.52.

EXPRESS specification:

```

*)
  ENTITY core_box_vent
    SUBTYPE OF (representation);
WHERE

(* ----- material_type ----- *)

wr1: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'material type') AND
  (it.description IN ['brass','nylon',
  'polyurethane','stainless steel'] )
  ))) =1;

(* ----- vent_type ----- *)

wr2: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'vent type')
  ))) =1;

(* ----- core_box_vent to equipment_size (as dimensions) ----- *)

wr3: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'diameter'))
  ))=1;

wr4: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'draft'))
  ))=1;

wr5: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'length'))
  ))=1;

wr6: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'width'))
  ))=1;

```

ISO 10303-223:2008 (E)

```
wr7: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'height'))
  ))=1;

(* ---- core_box_vent to shape_element (as core_shape) ---- *)

wr8: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION'
  IN TYPEOF(ap.rep_2)) ))=1;

END_ENTITY; -- core_box_vent
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'material type' and **description** of 'brass', 'nylon', 'polyurethane' or 'stainless steel'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'vent type'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'diameter'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'draft'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'length'.

**WR6:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'width'.

**WR7:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'height'.

**WR8:** Exactly one reference that is the **property** in exactly one **resource\_property\_representation** with a reference to **shape\_representation** through the **representation** attribute.

### 5.2.3.1.66 core\_conveyer\_system

A **core\_conveyer\_system** is a type of **equipment** that is used with the production of no bake or air set cores.

NOTE A **Core\_conveyer\_system** is defined according to 4.2.53.

#### EXPRESS specification:

```

*)
ENTITY core_conveyer_system
    SUBTYPE OF (core_equipment);

WHERE
    (* ----- cores_required----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'core required')) ) =1 ))))=1;

(* ----- cycle_time----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'cycle time')
    )))) <=1 ))))>=1;

(* ----- core_conveyer_system to equipment_size (as core_box_skid)----- *)

wr3:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'core box skid height')) ) <=1 ))))>=1;

```

ISO 10303-223:2008 (E)

```

wr4: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid length')) ) ) <=1 ))))>=1;

wr5: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid width')) ) ) <=1 ))))>=1;

wr6: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid diameter')) ) ) <=1 ))))>=1;

wr7: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid draft')) ) ) <=1 ))))>=1;

(* ----- core_conveyer_system to equipment_size (as core_plug)----- *)

wr8: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug height')) ) ) <=1 ))))>=1;

```

```

wr9: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug length')) ) ) <=1 ))))>=1;

wr10: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug width')) ) ) <=1 ))))>=1;

wr11: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug diameter')) ) ) <=1 ))))>=1;

wr12: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug draft')) ) ) <=1 ))))>=1;

END_ENTITY;
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_ - representation\_item** and **count\_measure** with a **name** of 'core required'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'cycle time'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box skid height'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box skid length'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box skid width'.

**WR7:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box skid diameter'.

**WR8:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box skid draft'.

**WR9:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core plug length'.

**WR10:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core plug width'.

**WR11:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core plug diameter'.

**WR12:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core plug draft'.

### 5.2.3.1.67 core\_equipment

A **core\_equipment** is a type of **casting\_equipment** that is used to produce cores or for core handling.

NOTE A **Core\_equipment** is defined according to 4.2.54.

#### EXPRESS specification:

\*)

```
ENTITY core_equipment
  SUBTYPE OF (casting_equipment);
END_ENTITY;
(*
```

### 5.2.3.1.68 core\_machine

A **core\_machine** is a type of **casting\_equipment** that is a machine that will force sand through an invest opening(s) to fill the core box cavity with core sand.

NOTE A **Core\_machine** is defined according to 4.2.55.



EXPRESS specification:

```

*)
  ENTITY core_machine
    SUBTYPE OF (core_equipment);
    WHERE
  (* ----- core_machine to core_box_vent (as vent definition)----- *)

  wr1:    SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.CORE_BOX_VENT' IN
    TYPEOF(pdr.representation))))=1;

  (* ----- core_machine ejector_equipment(as ejector_definition)--- *)

  wr2:    SIZEOF(QUERY ( arr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE') |
    (('CAST_PARTS_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP' IN TYPEOF(arr)) AND
    ('CAST_PARTS_SCHEMA.EJECTOR_EQUIPMENT' IN TYPEOF(arr.relying_resource)
    ))) = 1;

  (* ---core_machine to machine_mounting (as tool_mounting_definition)---- *)

  wr3:    SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.MACHINE_MOUNTING' IN
    TYPEOF(pdr.representation)) ))=1;
  END_ENTITY; -- core_machine
  (*

```

Formal propositions:

**WR1:** Exactly one reference that is the **resource** in exactly one **resource\_property** that is referenced through the **property** attribute of a **resource\_property\_representation** that references through the **representation** attribute a **representation** of type **core\_box\_vent**.

**WR2:** Exactly one reference through **usage** is an **action** that has exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **ejector\_system**.

**WR3:** Exactly one reference that is the **resource** in exactly one **resource\_property** that is referenced through the **property** attribute of a **resource\_property\_representation** that references through the **representation** attribute a **representation** of type **machine\_mounting**.

### 5.2.3.1.69 core\_master

A **core\_master** is a type of **core\_box\_tooling** that is constructed to create the production **Core\_box**.

NOTE A **Core\_master** is defined according to 4.2.56.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
  ENTITY core_master
    SUBTYPE OF (core_box_tooling);
    WHERE

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
  SELF) | (
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) )) = 1;

(* -----core box master_shrink_factor----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'core box master shrink factor')) )) =1 ))))=1;

  END_ENTITY; -- core_master
(*
```

Formal propositions:

**WR1:** The **die\_design\_feature** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'core box master shrink factor'.

### 5.2.3.1.70 core\_print

A **core\_print** is a type of **casting\_feature\_definition** that is attached to the core box designed to form a seat to locate and support a core within the mould.

NOTE A **Core\_print** is defined according to 4.2.57.

EXPRESS specification:

```

*)
ENTITY core_print
  SUBTYPE OF (sand_cast_design_feature);
WHERE

(* core_print to numeric_parameter (as clearance) *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'clearance')) = 1 ))) >=1;

(* core_print to drafted_surface (as draft) *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.DRAFTED_SURFACE' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='draft'))=1;

(* ----- core_print to item_size (as core_print_dimensions)----- *)

WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core print occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1;

(* ----- core_print to shape_element (as feature)----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core print occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 ))) >=1;
END_ENTITY; -- core_print
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'clearance'.

**WR2:** Exactly one **representation** used for the representation shall be of type **drafted\_surface**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'core print occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'core print size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'core print occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core print size usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.71 core\_setter

A **core\_setter** is a type of **core\_requirement** that is used with a specific moulding machine to set or place cores in the mould.

NOTE A **Core\_setter** is defined according to 4.2.58.

EXPRESS specification:

```

*)
ENTITY core_setter
  SUBTYPE OF (core_equipment);
  WHERE

(* ----- moulding_machine----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'moulding machine')
  )))) =1 ))))>=1;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'moulding machine'.

**5.2.3.1.72 corrosion\_service**

A **corrosion\_service\_requirement** is a type of **customer\_casting\_requirement** that is the type of enviromental or material conditions an in-service application will apply to the cast part.

NOTE A **Corrosion\_service\_requirement** is defined according to 4.2.59.

EXPRESS specification:

```

*)
ENTITY corrosion_service
    SUBTYPE OF (customer_casting_requirement);
    WHERE

(* ---- to descriptive_parameter(as corrosive_environment) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'corrosive environment')
    ) ) =1 )))=0));

(* ----- to material (as corrosive_material)----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'corrosive material')
    ) ) =1 )))=0));

(* ----- to numeric_parameter(as ph value) ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'ph value'))
    ) ) = 1 )))=0));

```

ISO 10303-223:2008 (E)

```
(* ---- to numeric_parameter(as temperature) ----- *)

wr4:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'temperature'))
  )) = 1 )))=0));
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'corrosive environment'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'corrosive material'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'ph value'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'temperature'.

### 5.2.3.1.73 customer\_casting\_requirement

A **customer\_casting\_requirement** is a type of **action\_property** that specify the necessary conditions that are to be executed or achieved for the production of the cast part.

NOTE A **Customer\_casting\_requirement** is defined according to 4.2.62.

#### EXPRESS specification:

```
*)
ENTITY      customer_casting_requirement
  SUBTYPE OF (property_definition);
WHERE

(* ----- new_tooling_required ----- *)

wr1:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'new tooling required') AND
  (it.description IN ['customers tool shop','foundry tool shop'] )
  ))) <=1 ))) = 0);
```

```

(* ----- specific_comments ----- *)

wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'specific comments')
  ))) >=0 )) ) ) = 0);

(* --- to illustration (as illustration)----- *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
  (adr.assigned_document)))>=0;

(* --- t to specification (as referenced_from)----- *)

wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'customer casting requirements' ) )
  >=0;

(* --- to shape_aspect (as applies_to)----- *)

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT'
  IN TYPEOF(pdr.related_property_definition.definition))
  ))>=0);

END_ENTITY;
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'new tooling required' and a **description** of either 'customers tool shop' or 'foundry tool shop'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'specific comments'.

**WR3:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **illustration**.

ISO 10303-223:2008 (E)

**WR4:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** in which the **kind** is exactly one **document\_kind** with **product\_data\_type** of 'customer casting requirement'.

**WR5:** Exactly one reference of **definition** is a **property\_process** that has one reference that is the **process** in exactly one **process\_property\_association** in which the **property\_or\_shape** is a **shape\_aspect**.

### 5.2.3.1.74 customer\_simulation

A **customer\_simulation** is a type of **simulation\_process** that is a request from a customer to use a simulation software tool to produce a simulation of a casting process for a proposed cast part.

NOTE A **Customer\_simulation** is defined according to 4.2.65.

#### EXPRESS specification:

\*)

```
ENTITY customer_simulation
  SUBTYPE OF (simulation_process);
  WHERE

wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc < *
  SELF\action_method_with_associated_documents.documents |
  (doc.kind.product_data_type='customer tool design report')))=1);

wr2: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc < *
  SELF\action_method_with_associated_documents.documents |
  (doc.kind.product_data_type='customer design deficiency report')
  ))=1);
END_ENTITY; -- customer_simulation
(*)
```

#### Formal propositions:

**WR1:** Each **customer\_simulation** shall be of type **action\_method\_with\_associated\_documents** that references through the **documents** attribute a **document** that references through the **kind** attribute a **document\_type** wthat has a **product\_data\_type** of 'customer tool design report'.

**WR2:** Each **customer\_simulation** shall be of type **action\_method\_with\_associated\_documents** that references through the **documents** attribute a **document** that references through the **kind** attribute a **document\_type** wthat has a **product\_data\_type** of 'customer design deficiency report'.



### 5.2.3.1.75 data\_curve

A **data\_curve** is a type of **compound\_representation\_item** that provides a set of data points for analyzing measurement values to determine expected results.

NOTE A **Data\_curve** is defined according to 4.2.67.

#### EXPRESS specification:

```

*)
ENTITY data_curve
    SUBTYPE OF (representation);
    WHERE

(* --- interpolation method----- *)

wr1: SIZEOF(QUERY( sri <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF (sri))
    AND (sri.name='interpolation method')))=1;

(* --- x unit----- *)

wr2: SIZEOF(QUERY( sri <* SELF.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (sri))
    AND (sri.name='x unit')))=1;

(* --- y unit----- *)

wr3: SIZEOF(QUERY( sri <* SELF.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (sri))
    AND (sri.name='y unit')))=1;

(* --- data_curve to curve_point (as data_points)----- *)
wr4: SIZEOF(QUERY( sri <* SELF.items |
    (('CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN TYPEOF (sri))
    ))>=1;

END_ENTITY; -- data_curve
(*)

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'interpolation method'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** with a **name** of 'x unit'.

ISO 10303-223:2008 (E)

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** with a **name** of 'y unit'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **cartesian\_point** with a **name** of 'data points'.

### 5.2.3.1.76 defect\_prediction

A **defect\_prediction** is a type of **action\_property** that is an undesirable structural feature that is forecast by a simulation.

NOTE A **Defect\_prediction** is defined according to 4.2.73.

#### EXPRESS specification:

```
*)
ENTITY defect_prediction
  SUBTYPE OF (action_method);
WHERE

  (* defect_prediction to location_element(as location) *)

wr1: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(mri))))=1)
  ))=1))))=0))))=0));

  (* defect_prediction to shape_aspect(as defect_shape) *)

wr2: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(eds))
  )) =1 )) ))=0));

END_ENTITY; -- defect_prediction
  (*
```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action\_property\_representation** through **property** that references through the **representation** attribute a **representation** that is of type **location\_shape\_representation**.

**WR2:** There shall be exactly one reference by an **action\_method\_assignment** through the **assigned\_action\_method** attribute that is of the type **applied\_action\_method\_assignment** and references through the **items** attribute a **shape\_aspect**.

### 5.2.3.1.77 defining\_action\_method\_relationship

A **defining\_action\_method\_relationship** is a kind of **action\_method\_relationship** in which the **related\_method** is an defining component.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.2.18 of ISO 10303-240:2005.

EXPRESS specification:

\*)

```
ENTITY defining_action_method_relationship
    SUBTYPE OF (action_method_relationship);
WHERE
    wr1: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY'
        IN TYPEOF(SELF.related_method));
END_ENTITY;
(*)
```

Formal propositions:

**WR1:** The **related\_method** shall be a **process\_plan\_activity**.

### 5.2.3.1.78 design\_part

A **design\_part** is a type of **product\_definition** that defines the cast part being manufactured.

NOTE A **Design\_part** is defined according to 4.2.76.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY design_part
  SUBTYPE OF (product_definition);
  WHERE

  wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ( 'CAST_PARTS_SCHEMA.INSTANCED_FEATURE' IN TYPEOF (sa)
    ))>=1;

  END_ENTITY;
  (*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape**, and it shall be referenced by exactly one **shape\_aspect** through the **of\_shape** attribute that is of type **instanced\_feature**.

### 5.2.3.1.79 design\_reference

A **design\_reference** is a type of **document** that is a graphics representation of a part.

NOTE 1 This application interpreted model is harmonized with the same object from paragraph 5.2.3.2.22 of ISO 10303-240:2005.

NOTE 2 A **Design\_reference** is defined according to 4.2.77.

EXPRESS specification:

```
*)
ENTITY design_reference
  SUBTYPE OF (document);
  WHERE

  wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    (duc.subject_element = 'drawing revision level')) = 1;

  wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
    ('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
    IN TYPEOF(dr)) |
    (NOT(SIZEOF(QUERY ( d <* adr.items |
    ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)
    )) >= 1)))) = 0;
```

```

wr3: SIZEOF(QUERY(dpa <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT') |
  (SIZEOF(['CAST_PARTS_SCHEMA.DESIGN_PART',
    'CAST_PARTS_SCHEMA.CAST_PART',
    'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
    'CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION']
    * TYPEOF(dpa.related_product))=1)
  )=1;

END_ENTITY;
(*)

```

### Formal propositions:

**WR1:** There shall be exactly one **document\_usage\_constraint** with **subject\_element** of 'drawing revision level' reference through the **source** attribute.

**WR2:** The **design\_reference** shall be the **assigned\_document** in exactly one **applied\_document\_reference** that contains one or more **process\_plan\_activity**.

**WR3:** there shall be exactly one **document\_product\_association** through the **relating\_document** attribute that references either a **cast\_part**, **cast\_part\_with\_rigging**, **design\_part** or **casting\_product\_definition** through the **related\_product** attribute.

### 5.2.3.1.80 die\_cast\_master

A **die\_cast\_master** is a type of **die\_cast\_tooling** that is a scale replica of the production tooling needed to produce cast part.

NOTE A **Die\_cast\_master** is defined according to 4.2.79.

### EXPRESS specification:

```

*)

ENTITY die_cast_master
  SUBTYPE OF (die_cast_tooling);
  WHERE

  (* -----master_shrink_factor----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'master shrink factor')) = 1 )))=1;

```

ISO 10303-223:2008 (E)

```
(* ***** to production_die_cast_mould (as female_component) ***** *)

wr3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die cast tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die cast mould female reference usage')
AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

(* ** die_cast_master to production_die_cast_mould (as male_component) *)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die cast tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die cast mould male reference usage')
AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

(* ** to machining_allowance (as master_allowance_on_master_tooling) * *)

wr5: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))=1);
END_ENTITY; -- die_cast_master
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'master shrink factor'.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'die cast tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die cast mould female reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_cast\_mould**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'die cast tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die cast mould male reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_cast\_mould**.

**WR5:** Exactly one reference by a **property\_definition** through the **definition** attribute and the **property\_definition** is of type **machine\_allowance**.

### 5.2.3.1.81 die\_cast\_tooling

A **die\_cast\_tooling** is a type of **casting\_product\_definition** that form a reusable mould to produce a near\_net shape precision cast part.

NOTE A Die\_cast\_tooling is defined according to 4.2.80.

#### EXPRESS specification:

\*)

```
ENTITY die_cast_tooling
  SUBTYPE OF (casting_product_definition);
  WHERE
```

```
(* ----- die_cast_tooling to material (as material_definition) *)
```

```
WR1: SIZEOF(QUERY (pdr <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='material_definition') AND
  (SIZEOF(QUERY (mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
  ))=1))))=1;
```

```
(* ***** to production_die_cast_mould (as die_equipment) ***** *)
```

```
wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='die cast tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die cast mould reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
  IN TYPEOF(sar.relating_shape_aspect))
  )) >=1 )))) >=1;
```

ISO 10303-223:2008 (E)

```
(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr3: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ))) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='tooling_location')))=1);

END_ENTITY; -- die_cast_tooling
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'material definition' and of type **make\_from\_usage\_option** in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'die cast tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die cast mould reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_cast\_mould**.

**WR3:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the items **attribute** that is of type **person\_and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_role** with **name** of 'tooling location'.

### **5.2.3.1.82 die\_clamping**

A **die\_clamping** is a type of **casting\_feature\_definition** that is the mechanism that secures the die mould to the diecasting machine.

NOTE A **Die\_clamping** is defined according to 4.2.81.



EXPRESS specification:

```

*)
ENTITY die_clamping
  SUBTYPE OF (die_design_feature);
WHERE

(* *** die_clamping to orientation (as die_clamping_positions) ***** *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'clamping_position')))) =1 )))=1;
END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **placement** with a **name** of 'clamping position'.

**5.2.3.1.83 die\_core**

A **die\_core** is a type of **casting\_production\_definition** that is a design of specially formed material inserted into a die mould to shape the interior or other part of a casting that cannot be easily shaped or formed by the die mould.

NOTE A **Die\_core** is defined according to 4.2.82.

EXPRESS specification:

```

*)
ENTITY die_core
  SUBTYPE OF (die_design_feature);
WHERE

(* ----- core_type----- *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'core type') AND
  ((it.description IN ['fixed core','hydraulic slide',
  'mechanical slide','sand core',
  'salt core','ceramic'] )) )) )) =1 )) )>=1;

```

ISO 10303-223:2008 (E)

```

(* ***** metal_type ***** *)

wr2: (SIZEOF(QUERY (co <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

(* die_core to drafted_surface (as draft) *)

wr3: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ))) =1;

(* ----- die_core to item_size (as guide_pin) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='die core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'guide pin size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect)
  )) >=0 )))) >=1;

(* ----- die_core to parting_surface (as parting) ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  ('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF(pdr.used_representation)) )) = 1;

(* ----- die_core to shape_aspect (as outside_shape_of_die_mould)----- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='die core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'die core area shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatinq_shape_aspect)
  )) =1 )))) >=1;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core type' and a **description** of either 'fixed core', 'hydraulic slide', 'mechanical slide', 'sand core', 'salt core' or 'ceramic'.

**WR2:** Exactly one reference that is the **definition** in exactly one **material\_designation**.

**WR3:** There shall be zero or more references that is a **representation** used for the implicit representation shall be of type **drafted\_surface**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'die core occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'guide pin usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR5:** There shall be zero or more references that is a **representation** used for the implicit representation shall be of type **parting\_surface**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'die core occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'die core area shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.84 die\_design\_feature

The **die\_design\_feature** is a type of **casting\_feature\_definition**, that defines design features on a die used to create a cast part.

NOTE A **Die\_design\_feature** is defined according to 4.2.83.

#### EXPRESS specification:

\*)

```
ENTITY die_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE
```

```
WR1: SIZEOF(QUERY( pdr <* get_property_definition_representations (SELF) |
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF ( pdr.used_representation ) ) ) =1;
```

ISO 10303-223:2008 (E)

```
(* ***** die_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 )))=1;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **die\_design\_feature** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **placement** with a **name** of 'orientation'.

### 5.2.3.1.85 die\_master

A **die\_master** is a type of **die\_cast\_tooling** that is a scaled replica of the production die needed to produce a cast part.

NOTE A **Die\_master** is defined according to 4.2.84.

EXPRESS specification:

```
*)
ENTITY die_master
  SUBTYPE OF (die_mould_tooling);
  WHERE

(* -----master_shrink_factor----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'master shrink factor')) =1 )))=1;
```

```

(* * die_master to production_die_cast_mould (as female_component) ** *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die master occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould female reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
  IN TYPEOF(sar.relater_shape_aspect))
  )) <=1 ))) >=1;

(* * die_master to production_die_cast_mould (as male_component) ***** *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die master occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould male reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
  IN TYPEOF(sar.relater_shape_aspect))
  )) <=1 ))) >=1;

(* * to machining_allowance (as machining_allowance_on_master_tooling) *)

wr5: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))>=0);
END_ENTITY; -- die_master
(*

```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'master shrink factor'.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'die master occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die mould female reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_mould**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'die master occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die mould male reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_mould**.

**WR5:** Exactly one reference by a **property\_definition** through the **definition** attribute and the **property\_definition** is of type **machine\_allowance**.

### 5.2.3.1.86 die\_mould\_tooling

A **die\_mould\_tooling** is a type of **casting\_product\_definition** that form a reusable mould to produce a near-net shape precision cast part.

NOTE A **Die\_mould\_tooling** is defined according to 4.2.86.

EXPRESS specification:

```

*)
ENTITY die_mould_tooling
    SUBTYPE OF (casting_product_definition);
    WHERE

(* ----- die_mould_tooling to material (as material_definition) *)

WR1: SIZEOF(QUERY (pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
    (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
    (pdr.name='material_definition') AND
    (SIZEOF(QUERY (mfuo <* USEDIN(pdr.related_product_definition,
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
    ))=1))))=1;

(* ** die_mould_tooling to production_die_mould (as die_equipment) *** *)

Wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='die mould tooling occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'production die mould reference usage') AND
    ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
    IN TYPEOF(sar.relying_shape_aspect))
    )) >=2 )))) >=1;

```

```

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

Wr3: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ))) >=1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='tooling_location'))>=1));
END_ENTITY; -- die_mould_tooling
(*

```

### Formal propositions:

**WR1:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'material definition' and of type **make\_from\_usage\_option** in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'die mould tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production die mould reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_die\_mould**.

**WR4:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the items **attribute** that is of type **person\_and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_role** with **name** of 'tooling location'.

### 5.2.3.1.87 die\_mould\_vent

A **die\_mould\_vent** is a type of **casting\_product\_definition** that is method by which trapped air can escape from a die cavity.

NOTE A **Die\_mould\_vent** is defined according to 4.2.87.

EXPRESS specification:

```

*)
ENTITY die_mould_vent
  SUBTYPE OF (die_design_feature);
WHERE

(* ***** die_mould_vent to ejector_pin (as ejector_pin_definition) *****
*)

wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die mould vent occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector pin reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PIN' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'die mould vent occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector pin reference usage' and of type **shape-defining\_relationship** in which the **relying\_shape\_aspect** is a **ejector\_pin**.

### 5.2.3.1.88 dimensional\_measurement\_inspection

A **dimensional\_measurement\_inspection** is a type of **inspection\_activity** that is a CMM procedure for evaluating the dimensional accuracy of a cast part against the design specification for every dimension with is associated tolerance for the part.

NOTE A **Dimensional\_measurement\_inspection** is defined according to 4.2.90.



EXPRESS specification:

```

*)
  ENTITY dimensional_measurement_inspection
    SUBTYPE OF (inspection_activity);
  WHERE

  (* -- to external_schema_definition (as iso10303_219_data) ----- *)

  wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(sar)) AND
    (SIZEOF(QUERY ( edi <* sar.items |
    ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA'
    IN TYPEOF(edi)) ) ) <= 1))))>=0;
  END_ENTITY;
  (*

```

Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the items attribute **externally\_defined\_schema**.

**5.2.3.1.89 directed\_dimensional\_location**

The **directed\_dimensional\_location** specifies is a type of **dimension\_location** that identifies the direction to measure the location dimension.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.11 of ISO 10303-224:2006.

EXPRESS specification:

```

*)
  ENTITY directed_dimensional_location
    SUBTYPE OF (dimensional_location);
  END_ENTITY; -- directed_dimensional_location
  (*

```

Attribute definitions:

**SELF\shape\_aspect\_relationship.relying\_shape\_aspect:** the origin of the directed dimension.

**SELF\shape\_aspect\_relationship.related\_shape\_aspect:** the target of the directed dimension.

### 5.2.3.1.90 document\_file

A **document\_file** is a type of **document** and **characterized\_object** that is the representation of the physical document that contains the information about cast parts specifications.

NOTE 1 A **Document\_file** is defined according to 4.2.95.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.31 of ISO 10303-240:2005.

#### EXPRESS specification:

\*)

```

ENTITY document_file
    SUBTYPE OF (document, characterized_object);
WHERE
    wr1 : SIZEOF(QUERY ( drt <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT') |
        (drt.name IN ['physical', 'digital']) )) = 1;

    wr2 : SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS') |
        (('CAST_PARTS_SCHEMA.IDENTIFICATION_ASSIGNMENT'
        IN TYPEOF(adr)) AND (adr.role.name = 'version')) )) <= 1;

    wr3: SIZEOF(QUERY ( aeia <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
        ('CAST_PARTS_SCHEMA.EXTERNAL_IDENTIFICATION_ASSIGNMENT'
        IN TYPEOF(aeia)) )) >= 0;

    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ((pd.name='document property') AND (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('CAST_PARTS_SCHEMA.DOCUMENT_FILE_PROPERTIES'
        IN TYPEOF(pdr.used_representation)) AND
        (pdr.used_representation.name = 'document format')))) <=1)))) <=1;

END_ENTITY;
(*)

```

Formal propositions:

**WR1:** The **document\_file** shall be the **represented\_file** in exactly one **document\_representation\_type** with **name** = 'physical' or 'digital'.

**WR2:** The **document\_file** shall be in the set of **items** of exactly one **applied\_identification\_assignment** with a **role** that defines an **identification\_role** with a **name** of 'version' that defines the **document\_file** version.

**WR3:** The **document\_file** shall be referenced by zero or more **applied\_external\_identification\_assignment** through the **items** attribute.

**WR4:** The **document\_file** shall be referenced by zero or one **property\_definition\_representation** that shall define a **document\_file\_properties** with a **name** of 'document format'.

### 5.2.3.1.91 document\_file\_properties

The **document\_file\_properties** is a type of **representation** that defines properties for **document\_file**.

NOTE 1 A **Document\_file\_properties** is defined according to 4.2.87.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.31 of ISO 10303-240:2005.

EXPRESS specification:

\*)

```
ENTITY document_file_properties
  SUBTYPE OF (representation);
  WHERE

  wr1: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='country code') AND
    ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))) )) <=1;

  wr2: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='detail level') AND
    ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))) )) <=1;

  wr3: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='geometry type') AND
    ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))) )) <=1;

  wr4: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='language code') AND
    ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))) )) <=1;
```

## ISO 10303-223:2008 (E)

```
wr5: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating interface') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr6: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating operating system') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr7: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating system') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr8: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='data format') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr9: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='format character code') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr10: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='file size') AND
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(r) = 2))) ) <=1;

wr11: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='page count') AND
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
  TYPEOF(r) = 2))) ) <=1;

END_ENTITY;
(*
```

### Formal propositions:

**WR1:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'country code'.

**WR2:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'detail level'.

**WR3:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'geometry type'.

**WR4:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'language code'.

**WR5:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'creating interface'.

**WR6:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'creating operating system'.

**WR7:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'creating system'.

**WR8:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'data format'.

**WR9:** There shall zero or one reference to a **descriptive\_representation\_item** with a **name** of 'format character code'.

**WR10:** There shall zero or one reference to a **measure\_representation\_item** and a **ratio\_measure\_with\_unit** with a **name** of 'file size'.

**WR11:** There shall zero or one reference to a **measure\_representation\_item** and a **count\_measure\_with\_unit** with a **name** of 'page count'.

### 5.2.3.1.92 drafted\_surface

A **draft\_surface** is a type of **shape\_representation** that represents geometry defined by a portion of an associated surface for the use as a draft surface as defined as the application object.

NOTE A **Drafted\_surface** is defined according to 4.2.98.

#### EXPRESS specification:

```

*)
ENTITY drafted_surface
  SUBTYPE OF (shape_representation);
  WHERE

  wr1: (SIZEOF(QUERY( it <* SELF.items |
    (SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'draft'))=1);

```

ISO 10303-223:2008 (E)

```
wr2: (SIZEOF(QUERY (rep <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (SIZEOF(['CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' ,
  'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'])
  * TYPEOF(rep.rep_2)=1)))=1);
```

```
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** One of the **representation\_items** used for the implicit representation of a **draft\_surface** shall be of type **measure\_representation\_item** and **plane\_angle\_measure\_with\_unit** with a **name** of 'draft'.

**WR2:** There shall be exactly one reference by an **representation\_relationship** through **rep\_1** that references through the **rep\_2** attribute a **representation** that is of type **planar\_shape\_representation** or **face\_shape\_representation**.

### 5.2.3.1.93 dryer

A **dryer** is a type of **core\_equipment** that is used to bake oil sand cores in an oven before they are placed in the mould.

NOTE A **Dryer** is defined according to 4.2.99.

#### EXPRESS specification:

```
*)
ENTITY dryer
  SUBTYPE OF (core_equipment);
  WHERE
  (* material_type *)

  wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'material type')
    )))) =1 ))))>=1;
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'material type'.

### 5.2.3.1.94 edge\_shape\_representation

An **edge\_shape\_representation** is a type of **representation** that represents geometry defined by a portion of an associated edge.

NOTE An **Edge\_shape\_element** is defined according to 4.2.100 and defines the requirement for **edge\_shape\_representation**.

EXPRESS specification:

```
*)
ENTITY edge_shape_representation
  SUBTYPE OF (shape_representation);
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **edge\_shape\_representation** shall have one or more **representation\_item** in its set of **items**.

**WR2:** The geometric element that is used to represent the **edge\_shape\_representation** shall be a **edge\_curve**, **oriented\_edge**.

### 5.2.3.1.95 ejector\_box

An **ejector\_box** is a type of **casting\_product\_definition** that is the frame that houses a function ejector system and is secured to the moving half core box or die tool.

NOTE A **Ejector\_box** is defined according to 4.2.101.

EXPRESS specification:

```
*)

ENTITY ejector_box
  SUBTYPE OF (ejector_design_feature);
  WHERE

  (* ***** ejector_box to ejector_plate (as plate_definition) ***** *)

wr1:  SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ejector box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector plate reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PLATE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;
```

ISO 10303-223:2008 (E)

```
(* ----- ejector_box to item_size (as box_size)----- *)

WR2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='ejector box occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'ejector box size usage') AND
('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
IN TYPEOF(sar.relatng_shape_aspect))
)) =1 ))) >=1;

(* ***** ejector_box to *)
(* ** production_core_box (as oriented_to_movable_die) ***** *)
(* ** production_die_mould (as oriented_to_movable_die) ***** *)
(* ** production_die_cast_mould (as oriented_to_movable_die) ***** *)
(* ** production_investment_cast_mould (as oriented_to_movable_die) * *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='ejector box occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'ejector box die reference usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX',
'CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD',
'CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD',
'CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD']
* TYPEOF(sar.relatng_shape_aspect)) =1))) =1 ))) >=1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'ejector box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector plate reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **ejector\_plate**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'ejector box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'ejector box size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.



**WR3:** Exactly one **shape\_aspect** with a **description** of 'ejector box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector box die reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_core\_box**, **production\_die\_mould**, **production\_die\_cast\_mould**, **production\_investment\_cast\_mould**.

### 5.2.3.1.96 ejector\_design\_feature

The **ejector\_design\_feature** is a type of **casting\_feature\_definition**, that defines ejector features on a cast part.

NOTE A **Ejector\_design\_feature** is defined according to 4.2.102.

#### EXPRESS specification:

```
*)
ENTITY ejector_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
  SELF) | (
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation) )) = 1;

  (* ***** ejector_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 ))))=1;
END_ENTITY; -- ejector_design_feature
(*
```

#### Formal propositions:

**WR1:** The **die\_design\_feature** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **placement** through the **used\_representation** that has a **name** of 'orientation'.

### 5.2.3.1.97 ejector\_equipment

An **ejector\_equipment** is a type of **equipment** that is used in producing the ejector system.

NOTE A **Ejector\_equipment** is defined according to 4.2.103.

#### EXPRESS specification:

```
*)
ENTITY ejector_equipment
  SUBTYPE OF (casting_equipment);
  WHERE

  wr1:(NOT (SIZEOF(QUERY( am <* SELF.usage |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
    (NOT(SIZEOF(QUERY ( ama <* USEDIN(am,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY ( eds <* ama.items |
    ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM' IN TYPEOF(eds))
    ))=1)))=0))))=0));

END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the items attribute an **ejector\_system**.

### 5.2.3.1.98 ejector\_pin

An **ejector\_pin** is a type of **casting\_feature\_definition** that is a definition of each pin used in the ejector\_system for size, shape, and location.

NOTE A **Ejector\_pin** is defined according to 4.2.104.

#### EXPRESS specification:

```
*)
ENTITY ejector_pin
  SUBTYPE OF (ejector_design_feature);
  WHERE
```

```
(* ----- ejector_pins to item_size (as pin_dimensions) ----- *)

WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ejector pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector pin dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relating_shape_aspect))
  )) =1 ))) >=1;

(* ----- ejector_pins to machining_feature (as thread) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ejector pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'thread reference usage') AND
  ('CAST_PARTS_SCHEMA.THREAD' IN TYPEOF(sar.relating_shape_aspect))
  )) =1 ))) >=1;

(* ----- ejector_pins to pin_tips (as tip_definition) ----- *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ejector pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pin tip reference usage') AND
  ('CAST_PARTS_SCHEMA.PIN_TIP' IN TYPEOF(sar.relating_shape_aspect))
  )) =1 ))) >=1;
END_ENTITY; -- ejector_pin
(*
```

### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'ejector pin dimension usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'thread reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **thread**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'pin tip reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **pin\_tip**.

### 5.2.3.1.99 ejector\_plate

An **ejector\_plate** is a type of **casting\_feature\_definition** that is a definition of the plate size, shape, and the pin locations required for the ejector system assembly.

NOTE A **Ejector\_plate** is defined according to 4.2.105.

#### EXPRESS specification:

```

*)
  ENTITY ejector_plate
  SUBTYPE OF (ejector_design_feature);
  WHERE

  (* ***** material_type ***** *)

  wr1: (SIZEOF(QUERY (co <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

  (* ----- ejector_plate to item_size (as pin_dimensions) ----- *)

  WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='ejector plate occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'ejector plate dimension usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) =1 )))) >=1;

  (* ejector_plate to numeric_parameter (as plate_height) *)

  wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'plate height')) )) =1 ))))=1;
  
```

```

(* --- ejector_plate to ejector_pin (as ejector_pin_definition) ----- *)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector pin reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PIN' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;

(* ----- ejector_plate to stop_pin (as stop_pin_definition) ----- *)

wr5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'stop pin reference usage') AND
  ('CAST_PARTS_SCHEMA.STOP_PIN' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* to reset_or_return_pins (as reset_or_return_pin_definition) *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'reset or push pin reference usage') AND
  ('CAST_PARTS_SCHEMA.RESET_OR_PUSH_PIN'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;
  END_ENTITY; -- ejector_plate
(*

```

### Formal propositions:

**WR1:** Exactly one reference that is the **definition** in exactly one **material\_designation**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'ejector plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'stop pin reference usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'plate height'.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'ejector plate shape occurrence' that is the **related\_shape\_aspect** in exactly one **casting\_feature\_component\_relationship** in with a **description** of 'ejector plate shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector pin reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **ejector\_pin**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'stop pin reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **stop\_pin**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'ejector pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'reset or push reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **reset\_or\_push\_pin**.

### 5.2.3.1.100 ejector\_system

An **ejector\_system** is a type of **casting\_feature\_definition** that is a definition of the pin and plate information required for the ejectors.

NOTE A **Ejector\_system** is defined according to 4.2.106.

#### EXPRESS specification:

\*)

```
ENTITY ejector_system
  SUBTYPE OF (ejector_design_feature);
  WHERE
```

```
(* ----- system_operated_by----- *)
```

```
WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'system operated by') AND
  (it.description IN ['hydraulics','spring loaded',
  'simple mechanical'] )
  )))) =1 ))))>=1;
```

```
(* ----- ejector_system to ejector_plates (as plate_definition) ----- *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector system occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector system reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PLATE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

END_ENTITY; -- ejector_system
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'system operated by' and a **description** of either 'hydraulics', 'spring loaded' or 'simple mechanical'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'ejector system occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector plate reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **ejector\_plate**.

### 5.2.3.1.101 elevated\_temperature\_service

An **elevated\_temperature\_service** is a type of **customer\_casting\_requirement** that is the designed temperature requirements that the cast part will be subjected to for in service applications.

NOTE A **Elevated\_temperature\_service** is defined according to 4.2.107.

#### EXPRESS specification:

```
*)
ENTITY elevated_temperature_service
  SUBTYPE OF (customer_casting_requirement);
  WHERE

(* ----- to descriptive_parameter(as service_descriptions) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELf ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'service description')
  ) ) ) =1 )))=0));
```

```
(* ----- to numeric_parameter(as maximum_temperature) ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
  (it.name = 'maximum temperature'))
  )) = 1 )))=0));

(* ----- to numeric_parameter(as minimum_cyclic_temperature) ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
  (it.name = 'minimum cyclic temperature'))
  )) = 1 )))=0));

(* ----- to numeric_parameter(as maximum_cyclic_temperature) ----- *)

wr4:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
  (it.name = 'maximum cyclic temperature'))
  )) = 1 )))=0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'service descriptions'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'maximum temperature'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'maximum cyclic temperature'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'minimum cyclic temperature'.



### 5.2.3.1.102 equipment\_setting\_record

An **equipment\_setting\_record** is a type of **activity\_result\_record** that is record of equipment operations for a casting process.

NOTE A **Equipment\_setting\_record** is defined according to 4.2.112

#### EXPRESS specification:

```

*)
ENTITY equipment_setting_record
  SUBTYPE OF (activity_result_record);
  WHERE

  (* ----- to property_parameter (as actual_machine_setting) -----*)

  wr1: (SIZEOF(QUERY( it <* SELF.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM']
    * TYPEOF(it)) = 1) AND
    (it.name = 'actual_parameter')))=1);

  (* ----- to equipment_usage (as recorded_usage) -----*)

  wr2: (SIZEOF(QUERY( rpr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.REPRESENTATION') |
    ('CAST_PARTS_SCHEMA.EQUIPMENT_USAGE'
    IN TYPEOF (rpr.property.resource))))=1);

  END_ENTITY;
  (*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with name of 'actual\_parameter'.

**WR2:** There shall be exactly one reference through the **used\_representaion** attribute by **property\_definiton\_representation** that is of type **equipment\_usage**.

### 5.2.3.1.103 equipment\_usage

An **equipment\_usage** is a type of **property\_definition** that is the context or circumstance in which an equipment is to be used and the parameter values for its use in that context or circumstance.

NOTE A **Equipment\_usage** is defined according to 4.2.115.

EXPRESS specification:

```

*)
ENTITY equipment_usage
  SUBTYPE OF (requirement_for_action_resource);
  WHERE

  (* ----- to equipment (as equipment_usage_context) -----*)

  wr1: (SIZEOF(QUERY( equ <* SELF.resources |
    ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(equ))))=1);

  (* ----- to property_parameter (as equipment_parameter) -----*)

  wr2: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    ('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
    * TYPEOF(it)) = 1)))) =1 )))) >=1;

  END_ENTITY;
  (*

```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action\_resource\_requirement** of type **requirement\_for\_action\_resource** through the **operations** that references one or more **equipment** through **resources**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item**.

**5.2.3.1.104 externally\_defined\_class**

An **externally\_defined\_class** is a type of **externally\_defined\_item** and a type of **class** that specifies a type of classification assignment with external reference.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.34 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY externally_defined_class
  SUBTYPE OF (externally_defined_item,class);
  WHERE
  wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

  wr2: SELF.source.name = 'ISO 13584 library';

```

```

(* class_BSU to supplier_BSU *)

wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF.source,
    'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    (aoa.role.name = 'library supplier'))) = 1;

(* class_BSU.version *)

wr4: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
    (('CAST_PARTS_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
    IN TYPEOF(aoa)) AND (aoa.role.name = 'class version')))) = 1;

END_ENTITY;
(*

```

#### Formal propositions:

**WR1:** The **source** attribute shall reference a **known\_source**.

**WR2:** The **source** attribute shall reference a **known\_source** with a **name** of 'ISO 13584 library'.

**WR3:** The **source** attribute shall reference a **known\_source** that is in the list of **items** for an **applied\_ - organization\_assignment** that has an **organization\_role** with a **name** of 'library supplier'.

**WR4:** The **externally\_defined\_class** shall be referenced by exactly one **library\_class\_version\_ - assignment** of kind **applied\_external\_identification\_assignment** through the **items** attribute with an **identification\_role** with **name** of 'class version'.

#### Informal proposition:

**IP1:** The value of **externally\_defined\_class** attribute **item\_id** (inherited from **externally\_defined\_item**) shall be instantiated in accordance to the class code of ISO 13584-42.

### **5.2.3.1.105 externally\_defined\_dimension\_definition**

An **externally\_defined\_dimension\_definition** is a type of **dimensional\_size** and a type of **externally\_ - defined\_item** that specifies a type of dimensional size with an external reference.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.12 of ISO 10303-224:2006.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY externally_defined_dimension_definition
  SUBTYPE OF (externally_defined_item, dimensional_size);
WHERE

  wr1: SELF.source.description =
    'externally defined dimension specification';

  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (adr.assigned_document.description =
    'externally defined size dimension specification')
    ) ) <= 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **source** attribute shall reference a **external\_reference** with a **description** of 'externally defined dimension specification'.

**WR2:** The **externally\_defined\_dimension\_definition** shall be reference by a through the **items** attribute by a **document\_reference** of type **applied\_document\_reference** that references through the **assigned\_document** attribute a **document** with **description** of 'externally defined dimension specification'.

### 5.2.3.1.106 externally\_defined\_general\_property

An **externally\_defined\_general\_property** is a type of **general\_property** and a type of **externally\_defined\_item** that specifies a type of general property with an external reference.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.15 of ISO 10303-224:2006.

EXPRESS specification:

```
*)
ENTITY externally_defined_general_property
  SUBTYPE OF (general_property, externally_defined_item);
WHERE
  wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

  wr2: SELF.source.name = 'ISO 13584 library';
```

```
(* property_BSU.version *)

wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
  IN TYPEOF(aoa)) AND
  (aoa.role.name = 'property version')))) = 1;

wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM') |
  ((ap.name = 'name scope') AND
  ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_CLASS' IN
  TYPEOF(ap.related_item)))) >= 1;

END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** The **source** attribute shall reference a **known\_source**.

**WR2:** The **source** attribute shall reference a **known\_source** with a **name** of 'ISO 13584 library'.

**WR3:** The **externally\_defined\_general\_property** shall be referenced by exactly one **library\_class\_version\_assignment** of kind **applied\_external\_identification\_assignment** through the **items** attribute with an **identification\_role** with **name** of 'property version'.

**WR4:** The **externally\_defined\_general\_property** shall be referenced by an **externally\_defined\_item\_relationship** with the **name** of 'name scope' through the **relating\_item** that references an **externally\_defined\_class** through the **related\_item**.

### 5.2.3.1.107 externally\_defined\_representation\_with\_parameters

An **externally\_defined\_representation\_with\_parameters** is a type of **representation** that defines placement and orientation for an external reference.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.15 of ISO 10303-224:2006.

#### EXPRESS specification:

```
*)
ENTITY externally_defined_representation_with_parameters
  SUBTYPE OF (representation);
WHERE

  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') |
    (('CAST_PARTS_SCHEMA.APPLIED_LIBRARY_ASSIGNMENT' IN TYPEOF(adr)) AND
    (adr.role.name='definitional class membership')))) = 1;
```

## ISO 10303-223:2008 (E)

```
wr2: SIZEOF(QUERY ( adr <* SELF.items |
  ('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(adr)) )) = 1;

wr3: SIZEOF(QUERY ( adr <* SELF.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(adr)) AND
  ('CAST_PARTS_SCHEMA.CARTESIAN_POINT'
  IN TYPEOF(adr.location))) )) = 1;
END_ENTITY;
(*
```

### Formal propositions:

**WR1:** The **externally\_defined\_representation\_with\_parameters** shall be referenced by exactly one **classificaton\_assignment** of kind **applied\_classification\_assignment** through the **items**.

**WR2:** There shall be zero or one **placement** in the list of **items**.

**WR3:** There shall be zero or one **placement** in the list of **items** with a **location** of **cartesian\_point**.

### 5.2.3.1.108 externally\_defined\_schema

The **externally\_defined\_schema** is a type of **externally\_defined\_item** that defines an external schema to be referenced for additional process planning information.

NOTE 1 A **Externally\_schema\_definition** is defined according to 4.2.118 and defines the requirements for externally\_defined\_schema.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.38 of ISO 10303-240:2005.

### EXPRESS specification:

```
*)
ENTITY externally_defined_schema
  SUBTYPE OF (externally_defined_item);
  WHERE
wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

wr2: SELF.source.name = 'ISO 10303 part';

wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.description = 'externally defined schema') ))
  <= 1;

wr4: SELF.item_id IN ['externally defined schema','workingstep',
  'nc function'];
END_ENTITY; -- externally_defined_schema
(*
```

Formal propositions:

**WR1:** The **source** attribute shall reference a **known\_source**.

**WR2:** The **source** attribute shall reference a **known\_source** with a **name** of 'ISO 10303 part'.

**WR3:** The **externally\_defined\_schema** shall be reference by a through the **items** attribute by a **document\_reference** of type **applied\_document\_reference** that references through the **assigned\_document** attribute a **document** with **description** of 'externally defined schema'.

**WR4:** The **item\_id** attribute shall be one of 'externally defined schema' or 'executable'.

### 5.2.3.1.109 fabrication

A **fabrication** is a type of **manufacturing\_activity** that identifies the activity of joining, usually by welding, of two or more parts to produce a finished assembly.

NOTE A **Fabrication** is defined according to 4.2.121.

EXPRESS specification:

\*)

```
ENTITY fabrication
  SUBTYPE OF (non_casting_activity);
END_ENTITY;
(*
```

### 5.2.3.1.110 face\_shape\_representation\_relationship

A **face\_shape\_representation\_relationship** is a type of **representation\_relationship** that is the representation of several **face\_shape\_representations** and their relationship to one another.

NOTE 1 A **Face\_shape\_element** is defined according to 4.2.122 and defines the requirements for **face\_shape\_representation**.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.16 of ISO 10303-224:2006.

EXPRESS specification:

\*)

```
ENTITY face_shape_representation_relationship
  SUBTYPE OF (representation_relationship);
WHERE

  wr1: ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(SELF.rep_1));
```

```
wr2: ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
      IN TYPEOF(SELF.rep_2));
```

```
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **face\_shape\_representation\_relationship** shall have only **face\_shape\_representation** in its **rep\_1**.

**WR2:** The **face\_shape\_representation\_relationship** shall have only **face\_shape\_representation** in its **rep\_2**.

### 5.2.3.1.111 feeder

A **feeder** is a type of **casting\_feature\_definition** that provides a channel for the molten metal to flow to the riser.

NOTE A **Feeder** is defined according to 4.2.125.

EXPRESS specification:

\*)

```
ENTITY feeder
SUBTYPE OF (gating_design_feature);
WHERE
```

```
(* feeders to connection_transition (as transition) *)
```

```
wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name IN['edge round'])) ) =1 ))))=1;
```

```
wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name IN [ 'fillet'])) ) =1 ))))=1;
```



```
(* ----- feeder to item_size (as shape) ----- *)

WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='feeder occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'feeder size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relater_shape_aspect))
  )) =1 )))) >=1;

(* ----- feeders to risers (as runner_connection) ----- *)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='feeder occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser reference usage') AND
  ('CAST_PARTS_SCHEMA.RISER' IN TYPEOF(sar.relater_shape_aspect))
  )) >=1 )))) >=1;
END_ENTITY; -- feeder
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'edge round'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'fillet'.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'feeder occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'feeder size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'feeder occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'riser reference usage' in which the **relating\_shape\_aspect** is a **riser**.

### 5.2.3.1.112 filter

A **filter** is a type of **casting\_feature\_definition** that prevents the entrance of slag and other materials from entering the mould.

NOTE A **Filter** is defined according to 4.2.126.

EXPRESS specification:

```

*)
ENTITY filter
  SUBTYPE OF (gating_design_feature);
  WHERE

(* ----- type_of_filter----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'filter type')))) =1 )))=1;

(* ----- filter to item_size (as filter_shape) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='filer occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'filter size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1;
END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with name of 'filter type'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'filter occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'filter size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.113 finishing\_and\_machining\_operation

A **finishing\_and\_machining\_operation** is a type of **customer\_casting\_requirement** that are value added operations to the casting(s) before delivery of the cast part(s) to the customer.

NOTE A **Finishing\_and\_machining\_operation** is defined according to 4.2.128.

#### EXPRESS specification:

```

*)
ENTITY finishing_and_machining_operation
    SUBTYPE OF (customer_casting_requirement);
    WHERE

(* ----- to descriptive_parameter(as comments) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'comments')
    ) ) =1 )))=0));

(* ----- to machining_activity (as finishing_operations) ----- *)

wr2:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
    ((aam\action_method_assignment.role.name='finishing operation') AND
    ('CAST_PARTS_SCHEMA.MACHINING_ACTIVITY'
    IN TYPEOF(aam\action_method_assignment.assigned_action_method))))=1);

(* ----- to machining_activity (as machining_operations) ----- *)

wr3:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
    ((aam\action_method_assignment.role.name='machining operation') AND
    ('CAST_PARTS_SCHEMA.MACHINING_ACTIVITY'
    IN TYPEOF(aam\action_method_assignment.assigned_action_method))))=1);
END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'comments'.

**WR2:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** that references through the role attribute an **action\_method\_role** with a name of 'finishing operation' and the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** that references through the items attribute **machining\_activity**.

**WR3:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** that references through the role attribute an **action\_method\_role** with a name of 'machining operation' and the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** that references through the items attribute **machining\_activity**.

### 5.2.3.1.114 finishing\_equipment

A **finishing\_equipment** is a type of **casting\_equipment** that is the equipment to provide additional operations on the cast part.

NOTE A **Finishing** is defined according to 4.2.127 and defines the requirements for **finishing\_equipment**.

EXPRESS specification:

\*)

```
ENTITY finishing_equipment
    SUBTYPE OF (casting_equipment);
END_ENTITY;
(*)
```

### 5.2.3.1.115 first\_article

A **first\_article** is a type of **casting\_verification** that is the first group of cast parts that are produced from tooling for the first time and the tooling has not been certified for production usage.

NOTE A **First\_article** is defined according to 4.2.129.

EXPRESS specification:

```

*)

ENTITY first_article
  SUBTYPE OF (casting_verification);
  WHERE

  (* ----- number of pieces -----*)

wr1: (NOT (SIZEOF (QUERY (ars <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
  (NOT (SIZEOF (QUERY (ap <* USEDIN (ars.method,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of pieces'))
  )) =1 ))) =0 )))=0 )))=0)));

  (* --- first_article to approval (as customer_approval) ----- *)

wr2: SIZEOF (QUERY (aaa <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |
  (NOT (SIZEOF (QUERY (ra <* USEDIN (aaa,
  'CAST_PARTS_SCHEMA.ROLE_ASSOCIATION.ITEM_WITH_ROLE') |
  (ra.role.name='customer approval'))=1 )))=0;

  (* ----- first_article to date (as delivery_date) ----- *)

wr3: (SIZEOF (QUERY (ada <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (ada.role.name='delivery date'))=1);

  (* --first_article to master_sample (as customer_approved) ----- *)

wr4: SIZEOF (QUERY (ada <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT (SIZEOF (QUERY (ari <* ada.items |
  'CAST_PARTS_SCHEMA.MASTER_SAMPLE' IN TYPEOF(ari))) =1))))=0;

  END_ENTITY; -- first_article
  (*

```

Formal propositions:

**WR1:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **count\_measure** with a **name** of 'number of pieces'.

**WR2:** There shall be exactly one reference by a **applied\_approval\_assignment** through **items** that references **date\_assignment** with a **name** of 'foundary approval' through role, and is of kind **approval\_assignment** that references a **approval** through **assigned\_approval**.

**WR3:** There shall be exactly one reference by an **applied\_date\_assignment** through the **items** attribute that is of type **date\_assignment** that references through the **role** attribute a **date\_role** with **name** of 'delivery date'.

**WR4:** There shall be exactly one reference by an **action\_request\_assignment** through the **assigned\_action\_request** attribute that references through the **items** attribute a **master\_sample**.

### 5.2.3.1.116 **fixture\_tab**

A **fixture\_tab** is a type of **casting\_design\_feature** that is an appendage added to the casting to secure or locate the cast part for machining.

NOTE A **Fixture\_tab** is defined according to 4.2.130.

EXPRESS specification:

\*)

```
ENTITY fixture_tab
  SUBTYPE OF (casting_design_feature);
END_ENTITY;
(*)
```

### 5.2.3.1.117 **flask**

A **flask** is a type of **casting\_product\_definition** that is typically wooden or metal, that consists of two or more sections used to enclose the material in which the mould is formed.

NOTE A **Flask** is defined according to 4.2.131.

EXPRESS specification:

```

*)
ENTITY flask
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- flask to descriptive_parameter (as type_of_flask) ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'flask type')))) =1 ))))=1;

(* ----- flask to numeric_parameter (as cope_height) ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'cope height')) ) =1 ))))=1;

(* ----- flask to numeric_parameter (as cope_height) ----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'drag height')) ) =1 ))))=1;

(* ----- flask to numeric_parameter (as flask_length) ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'flask length')) ) =1 ))))=1;

```

ISO 10303-223:2008 (E)

```
(* ----- flask to numeric_parameter (as flask width) ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
   TYPEOF(pdr.used_representation)) AND
   (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
              'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND
              (it.name = 'flask width')) ) =1 )))=1;

(* ----- flask to pin_center (as pin_center_definition) ----- *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='flask reference occurrence') AND
   (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'pin center reference usage') AND
    ('CAST_PARTS_SCHEMA.PIN_CENTER' IN TYPEOF(sar.relater_shape_aspect))
    )) <=1 ))) >=1;

(* ----- flask to sand_mould (as generates) ----- *)

wr7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='flask reference occurrence') AND
   (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'sand mould reference usage') AND
    ('CAST_PARTS_SCHEMA.SAND_MOULD' IN TYPEOF(sar.relater_shape_aspect))
    )) =1 ))) >=1;

END_ENTITY; -- flask
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'flask type'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'cope height'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'drag height'.



**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'flask length'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'flask width'.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'flask occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'pin center reference usage' in which the **relating\_shape\_aspect** is a **pin\_center**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'flask occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'sand mould reference usage' in which the **relating\_shape\_aspect** is a **sand\_mould**.

### 5.2.3.1.118 flaskless

A **flaskless** is a type of **action\_property** that may require a removable flask jacket slipped around the mould when the mould is filled and may also require mould weights to be placed on top of the mould before filling the mould.

NOTE A **Flaskless** is defined according to 4.2.132.

#### EXPRESS specification:

\*)

ENTITY flaskless

    SUBTYPE OF (sand\_cast\_design\_feature);

    WHERE

(\* ----- description ----- \*)

```
wr1:  SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'description'))))=1 )))=1;
```

## ISO 10303-223:2008 (E)

```
(* ----- flaskless to mould_box (as mould_made_by) ----- *)

Wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='flaskless reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'mould box reference usage') AND
  ('CAST_PARTS_SCHEMA.MOULD_BOX' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;
END_ENTITY; -- flaskless
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'description'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'flaskless occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'mould box reference usage' in which the **relating\_shape\_aspect** is a **mould\_box**.

### 5.2.3.1.119 gating\_design\_feature

The **gating\_design\_feature** is a type of **casting\_feature\_definition**, that defines gating features on a cast part.

NOTE A **Gating\_design\_feature** is defined according to 4.2.133.

### EXPRESS specification:

\*)

```
ENTITY gating_design_feature
SUBTYPE OF (casting_feature_definition);
WHERE
```

```
wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
  SELF) | (
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) )) = 1;
```

```
(* ----- to Orientation (as placement) ----- *)

wr2:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 )))=1;
END_ENTITY;
(*
```

### Formal propositions:

**WR1:** The **die\_design\_feature** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **placement** through the **used\_representation** that has a **name** of 'orientation'.

### 5.2.3.1.120 gating\_system

A **gating\_system** is a type of **casting\_feature\_definition** that a network of channels connected to a funnel like shape at the top of the mould through which molten metal flows into the mould cavity to produce the cast part.

NOTE A **Gating\_system** is defined according to 4.2.134.

#### EXPRESS specification:

```
*)
ENTITY gating_system
  SUBTYPE OF (gating_design_feature);
  WHERE

(* ----- gating_system to ingates (as gating_components) ----- *)

wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='gating system occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gating system components usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.INGATE',
  'CAST_PARTS_SCHEMA.RISER',
  'CAST_PARTS_SCHEMA.RUNNER',
  'CAST_PARTS_SCHEMA.SPRUE']
  * TYPEOF(sar.relatng_shape_aspect)) =1))) >=1 ))))>=1;

END_ENTITY; -- gating_system
(*
```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'gating system occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'gating system reference usage' in which the **relating\_shape\_aspect** is an **ingate**, **riser**, **runner** or **sprue**.

### 5.2.3.1.121 generic\_manufacturing\_resource

A **generic\_manufacturing\_resource** is a type of **action\_resource** that defines resources for process planning activities.

NOTE 1 A **Generic\_casting\_resource** is defined according to 4.2.136 and defines the requirements for **generic\_manufacturing\_resource**.

NOTE 2 This application interpreted model is harmonized with the entity **generic\_manufacturing\_resource** from paragraph 5.2.3.1.52 of ISO 10303-240:2005.

EXPRESS specification:

\*)

```
ENTITY generic_manufacturing_resource
    SUBTYPE OF (action_resource);
```

WHERE

```
(* -----quantity -----*)
```

```
wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'resource quantity')) ) =1 ))))>=1;
```

```
(* -----unit -----*)
```

```
wr2: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'resource quantity')AND
    ('CAST_PARTS_SCHEMA.CONTEXT_DEPENDENT_UNIT'
    IN TYPEOF(it\measure_with_unit.unit_component)) AND
    (it\measure_with_unit.unit_component\conversion_based_unit.name='each')
    ) ) =1 ) ) ) ) >=1;
```

```
END_ENTITY;
```

```
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **count\_measure\_with\_unit** with a **name** of 'resource quantity'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **count\_measure\_with\_unit** with a **name** of 'each'.

**5.2.3.1.122 heat\_treat**

A **heat\_treat** is a type of **manufacturing\_activity** that identifies the activity of heat treating the cast part.

NOTE A **Heat\_treat** is defined according to 4.2.144.

EXPRESS specification:

\*)

```

ENTITY heat_treat
  SUBTYPE OF (non_casting_activity);
  WHERE
wr1: (NOT (SIZEOF (QUERY (am <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF (am)) AND
  (SIZEOF (QUERY (cp <* am.items |
  ('CAST_PARTS_SCHEMA.HEAT_TREAT_REQUIREMENT'
  IN TYPEOF (cp)))) = 1 ))))=0));

  END_ENTITY;
(*)

```

Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_-assignment** of type **applied\_action\_method\_assignment** that references through the items attribute a **heat\_treat\_requirement**.

**5.2.3.1.123 heat\_treat\_requirement**

The **heat\_treat\_requirement** is a type of **process\_requirement** that is a sequence of heating and cooling cycles used to obtain desired properties or conditions of a cast part.

NOTE A **Heat\_treat\_requirement** is defined according to 4.2.145.

EXPRESS specification:

```

*)
ENTITY heat_treat_requirement
    SUBTYPE OF (process_requirement);
    WHERE

(* ----- additional_requirements ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'additional requirement')
    ) ) =1 )))=0));

(* ----- softening_anneal_required ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'softening anneal required') AND
    (it.description IN (['TRUE','FALSE']))
    ) ) =1 )))=0));

(* ----- stress_relief_required ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'stress relief required') AND
    (it.description IN (['TRUE','FALSE']))
    ) ) =1 )))=0));

(* --- to organization (as where_heat_treat_is_to_be_performed----- *)

wr4:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    ((aam\organization_assignment.role.name='activity organization id') AND
    ('CAST_PARTS_SCHEMA.ORGANIZATION'
    IN TYPEOF(aam\organization_assignment.assigned_organization))))=1);

(* ----- to process_property (as heat_treatment_per_design_data) ----- *)

wr5:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION'
    IN TYPEOF(pdr.related_property_definition)) AND
    (pdr.related_property_definition.name='process property'))
    )=1);
    
```

```
(* -----to specification (as heat_treat_specification) ----- *)

wr6:SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'heat_treat_specification') ))
  =1;

(* ---- to mechanical_property_requirements (as mechanical_property) ---- *)

wr7:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.MECHANICAL_PROPERTY_REQUIREMENTS'
  IN TYPEOF(pdr.related_property_definition)
  ))=1);

END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'additional requirements'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'softening anneal required' and a **description** of either 'true' or 'false'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'stress relief required' and a **description** of either 'true' or 'false'.

**WR4:** There shall be exactly one reference by an **applied\_organization\_assignment** through the **items** attribute that is of type **organization\_assignment** that references through the **role** attribute a **organization\_role** with **name** of 'activity organization id'.

**WR5:** Exactly one reference of **definition** is a **property\_process** that has one reference that is the **process** in exactly one **process\_property\_association** in which the **property\_or\_shape** is a **property\_definition** with **name** of 'process property'.

**WR6:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document**.

**WR7:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** in which the **kind** is exactly one **document\_kind** with **product\_data\_type** of 'heat treat user defined requirements'.

### 5.2.3.1.124 illustration

An **illustration** is a type of **document** that is a pictorial presentation of a part.

NOTE 1 An **illustration** is defined according to 4.2.147.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.53 of ISO 10303-240:2005.

#### EXPRESS specification:

\*)

```
ENTITY illustration
    SUBTYPE OF (document);
WHERE
```

```
wr1: SIZEOF(QUERY ( aduc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS') |
    ('CAST_PARTS_SCHEMA.VIEW_REFERENCE'
    IN TYPEOF
    (aduc\document_usage_constraint_assignment.assigned_document_usage))
    )) >= 0;
```

```
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** An **illustration** shall be referenced by one or more **document\_usage\_constraint** of kind **view\_reference**.

### 5.2.3.1.125 impregnation\_activity

An **impregnation\_activity** is a type of **manufacturing\_activity** that identifies the activity of sealing castings, ferrous and nonferrous, where microporosity is present.

NOTE An **Impregnation\_activity** is defined according to 4.2.148.

#### EXPRESS specification:

\*)

```
ENTITY impregnation_activity
    SUBTYPE OF (non_casting_activity);
END_ENTITY;
(*
```



### 5.2.3.1.126 in\_facility\_location

The **in\_facility\_location** is a type of **resource\_property** that defines the position of casting equipment.

NOTE 1 An **In\_facility\_location** is defined according to 4.2.149.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.43 of ISO 10303-240:2005.

#### EXPRESS specification:

\*)

```
ENTITY in_facility_location
  SUBTYPE OF (representation);
WHERE

  wr1: (NOT(SIZEOF(QUERY ( it <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'building or area')) )) =0));

  wr2: (NOT(SIZEOF(QUERY ( it <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'location code')) )) =0));

  wr3: (NOT(SIZEOF(QUERY ( it <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'sublocation')) )) =0));

END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'building or area'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'location code'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'sublocation'.

### 5.2.3.1.127 ingate

An **ingate** is a type of **casting\_feature\_definition** that connect the mould cavity, made by the pattern, to a runner through which molten metal flows.

NOTE An **Ingate** is defined according to 4.2.152.

EXPRESS specification:

```

*)
ENTITY ingate
  SUBTYPE OF (gating_design_feature);
  WHERE

(* ----- cope_or_drag ----- *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
  ((it.description IN ['cope', 'drag', 'cope and drag']
  )) )) )) =1 )) ))>=1;

(* ----- ingates to ingate_contacts (as contacts) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ingate contact reference usage') AND
  ('CAST_PARTS_SCHEMA.INGATE_CONTACT'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) >=1 )))) >=1;

(* ----- ingates to item_size (as ingate_shape) ----- *)

WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ingate size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) <=1 )))) >=1;
END_ENTITY; -- ingate
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'system operated by' and a **description** of either 'cope', 'drag' or 'cope and drag'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'ingate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'ingate contact reference usage' in which the **relating\_shape\_aspect** is an **ingate\_contact**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'ingate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'ingate size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.128 ingate\_contact

An **ingate\_contact** is a type of **casting\_feature\_definition** that is the connection information for each ingate at the point of contact with the production pattern and with the runner bar.

NOTE An **Ingate\_contact** is defined according to 4.2.151.

EXPRESS specification:

\*)

```
ENTITY ingate_contact
  SUBTYPE OF (gating_design_feature);
  WHERE
```

```
(* ----- cope_or_drag ----- *)
```

```
WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
  ((it.description IN ['cope', 'drag', 'cope and drag']
  )) )) )) =1 )) ))>=1;
```

ISO 10303-223:2008 (E)

```
(* - ingate_contacts to connection_transition (as runner_transition) -- *)

wr2 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'runner transition edge round')) ) ) <=1 ))))=1;

wr3 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'runner transition fillet')) ) ) <=1 ))))=1;

(* - ingate_contacts to connection_transition (as pattern_transition) - *)

wr4 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'pattern transition edge round')) ) ) <=1 ))))=1;

wr5 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'pattern transition edge fillet')) ) ) <=1 ))))=1;

(* -ingate_contacts to production_pattern_definition (as applied_to) - *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate contact occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production pattern definition reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;
```

```
(* ---- ingates_contacts to runner (as runner_connection) ----- *)

wr7:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate contact occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'runner reference usage') AND
  ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;

(* -- ingate_contacts to shape_element (as break_off_connection)----- *)

WR8:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate contact occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ingate contact shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;
  END_ENTITY; -- ingate_contact
  (*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'system operated by' and a **description** of either 'cope', 'drag' or 'cope and drag'.

**WR2:** Exactly one **representation** with **description** of 'runner transition' has exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'runner transition edge round'.

**WR3:** Exactly one **representation** with **description** of 'runner transition' has exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'runner transition fillet'.

**WR4:** Exactly one **representation** with **description** of 'pattern transition' has exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'pattern transition edge round'.

**WR5:** Exactly one **representation** with **description** of 'pattern transition' has exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'pattern transition fillet'.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'ingate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'production pattern definition reference usage' in which the **relating\_shape\_aspect** is a **production\_pattern\_definition**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'ingate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'runner reference usage' in which the **relating\_shape\_aspect** is a **runner**.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'ingate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'ingate contact shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.129 inspection\_activity

An **inspection\_activity** is a type of **process\_plan\_activity** that is an action, procedure, or activity used to evaluate, check, inspect, or validate a cast part or tooling used to produce a cast part.

NOTE An **Inspection\_activity** is defined according to 4.2.153.

EXPRESS specification:

\*)

```
ENTITY inspection_activity
    SUBTYPE OF (casting_activity);
    WHERE

    (* ----- inspection_level ----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'inspection_level')) ) =1 )))=1;
```

```

(* ----- reason_for_inspection ----- *)

wr2:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'reason for inspection') AND
        (it.description IN [
          'initial inspectrion',
          'engineering change',
          'tooling replacement',
          'tooling transfer',
          'tooling refurbishment',
          'discrepancy correction',
          'material or supplier source change',
          'production sample set'])))) =1 ))))>=1;

(* ----- type_of_inspection ----- *)

wr3:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'type of inspection') AND
        (it.description IN [
          'destructive',
          'dimensional',
          'functional testing',
          'gauging',
          'hardness property',
          'internal property',
          'leak testing',
          'material composition',
          'surface','visual'])))) =1 ))))>=1;

(* -----nspection_activity to approval (as foundry approval) ----- *)

wr4: (NOT(SIZEOF(QUERY(aaa <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |
  (SIZEOF(QUERY(ra <* USEDIN(aaa,
  'CAST_PARTS_SCHEMA.ROLE_ASSOCIATION.ITEM_WITH_ROLE') |
  (ra.role.name='foundry approval')) =1 )) )=0));

(* -- to casting_verification (as quality_assurance) ----- *)

wr5: SIZEOF(QUERY(aaa <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
  ('CAST_PARTS_SCHEMA.CASTING_VERIFICATION'
  IN TYPEOF(aaa.request)))=1;

```

ISO 10303-223:2008 (E)

```
(* --- to part_version (as inspected_part) ----- *)

wr6: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(am)) AND
(SIZEOF(QUERY( cp <* am.items |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(cp)))) =1 ))))=0));

(* -- to product_quality_report (as inspection_report) ----- *)

wr7: (('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
IN TYPEOF(SELF)) AND
(NOT (SIZEOF(QUERY( doc <* SELF.documents |
(doc.kind.product_data_type='product quality report'))=0)));

(* ----- to inspection_plan (as customer_requirements) ----- *)

wr8: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
(SIZEOF(QUERY( ip <* USEDIN(ap.related_method,
'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD') |
'CAST_PARTS_SCHEMA.INSPECTION_PLAN' IN TYPEOF(ip))=1)))=0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **count\_measure** with a **name** of 'inspection level'.

**WR2:** Exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** in which the **representation** is a **representation** in which exactly one **items** is a **descriptive\_representation\_item** with a **name** of 'reason for inspection' and **description** of 'initial inspection', 'engineering change', 'tooling replacement', 'tooling transfer', 'tooling refurbishment', 'discrepancy correction', 'material or supplier source change'.

**WR3:** Exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** in which the **representation** is a **representation** in which exactly one **items** is a **descriptive\_representation\_item** with a **name** of 'type of inspection' and **description** of 'destructive', 'dimensional', 'functional testing', 'gauging', 'hardness property', 'internal property', 'leak testing', 'material composition', 'surface', 'visual'.



**WR4:** Exactly one reference that is the **items** in exactly one **applied\_approval\_assignment** that is of type **approval\_assignment** that has a role attribute that references an **object\_role** with name of 'foundry approval'.

**WR5:** Exactly one referene is by an **action\_request\_solution** through the **method** attribute that references through the **request** attribute a **casting\_verification**.

**WR6:** Exactly one reference is by an **action\_method\_assignment** through the **assigned\_action\_method** that is of the **applied\_action\_method\_assignment** that references through the **items** attribute a **product\_definition**.

**WR7:** This entity is of type **action\_method\_with\_associated\_documents** that references through the **documents** attribute a **document** that references through the **kind** attribue a **document\_type** with **product\_data\_type** of 'product quality report'.

**WR8:** Exactly one reference is by an **action\_method\_relationship** through the **relating\_method** attribute that reference through the **related\_method** attribute an **action\_method** that is referenced by an **action** through the **chosen\_method** attribute that is of type **inspection\_plan**.

### 5.2.3.1.130 inspection\_or\_test\_requirement

An **inspection\_or\_test\_requirement** is a type of **casting\_requirement** that definies the requested inspections or test to be performed on a cast part.

NOTE An **Inspection\_or\_test\_requirement** is defined according to 4.2.154.

#### EXPRESS specification:

\*)

```
ENTITY inspection_or_test_requirement
    SUBTYPE OF (customer_casting_requirement);
WHERE
```

```
(* ----- validation_id ----- *)
```

```
wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'validation id')
    ) ) =1 )))=0));
```

```
(* ----- test name ----- *)
```

```
wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'test name')
    ) )=1 ))) = 0);
```

ISO 10303-223:2008 (E)

```
(* ----- to numeric_parameter(as frequency)----- *)

wr3:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'frequency'))=1))))=0);

(* ----- to numeric_parameter(as sampling_run_size)----- *)

wr4:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'sampling_run_size'))=1))))=0);

(* ----- to customer_casting_requirement (as inspection_against)----- *)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT'
  IN TYPEOF(pdr.related_property_definition)) ) ) = 1;

(* ---- to specification (as production_quality_requirement_standard)--- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.description =
  'production quality requirement standard') ) )
= 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'validation id'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'test name'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'frequency'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'sampe run size'.

**WR5:** Exactly one reference that is the **relating\_action\_property** in exactly one **action\_property\_relationship** in which the **related\_action\_property** is a **customer\_casting\_requirement**.

**WR6:** Exactly one reference that is the **items** attribute by a **document\_reference** of type **applied\_document\_reference** that references through the **assigned\_document** attribute a **document** with **description** of 'production quality requirement standard'.

### 5.2.3.1.131 inspection\_or\_test\_result

An **inspection\_or\_test\_result** is a type of **property\_definition** that is the outcome of an examination or measurement.

NOTE An **Inspection\_or\_test\_result** is defined according to 4.2.155.

#### EXPRESS specification:

```
*)
ENTITY inspection_or_test_result
  SUBTYPE OF (property_definition);
  WHERE

(* ----- result ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'result')
  ) )=1 )))) = 0);

(* -- inspection_or_test_result to checking_aid_tool (as aid_used) --- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CHECKING_AID_TOOL'
  IN TYPEOF(pdr.related_property_definition)
  ))=1);
```

## ISO 10303-223:2008 (E)

```
(* --inspection_or_test_result to shape_aspect(as applies_to)----- *)

wr3: (NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ( ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE'
  IN TYPEOF(pdr.related_property_definition)) AND
  (SIZEOF(QUERY( sa <* USEDIN(pdr.related_property_definition,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sa))
  )) =1 )) )) =0));

(* -- inspection_or_test_result to sampled_set(as belongs_to)----- *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.SAMPLED_SET'
  IN TYPEOF(pdr.related_property_definition))
  )) =1);

END_ENTITY;
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'inspection or test result'.

**WR2:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_definition\_relationship** in which the **relating\_property\_definition** is an **checking\_aid\_tool**.

**WR3:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_definition\_relationship** in which the **relating\_property\_definition** is an **shape\_aspect** that has exactly one reference through the **of\_shape** that is a **shape\_aspect**.

### **5.2.3.1.132 inspection\_plan**

An **inspection\_plan** is a type of **product\_definition\_process** that is a scheme of predetermined processes or procedures employed to evaluate the quality and dimensional accuracy of a cast part against the design record.

NOTE An **Inspection\_plan** is defined according to 4.2.156.

EXPRESS specification:

```

*)

ENTITY inspection_plan
  SUBTYPE OF (product_definition_process);
  WHERE

(* ----- to descriptive_parameter(as corrective_action) ----- *)

wr1:SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'test name') AND
  (it.description IN ['weld', 'impregnation']))) =1 ))))>=1;

(* -----to descriptive_parameter(as correction_procedures) ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'correction procedures') AND
  (it.description IN ['dimensional','visual','internal']))) =1 ))))>=1;

(* ----- to inspection_or_test_requirement(as customer_requirements) ----- *)

wr3:(SIZEOF(QUERY(ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
  ('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_REQUIREMENT'
  IN TYPEOF(ap.property_or_shape)
  ))=1);

(* ----- to special_inspection_requirement(as customer_requirements) ----- *)

wr4:(SIZEOF(QUERY(ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
  ('CAST_PARTS_SCHEMA.SPECIAL_INSPECTION_REQUIREMENT'
  IN TYPEOF(ap.property_or_shape)
  ))=1);

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'chemical analysis certification' and a **description** of either 'weld' or 'impregnation'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'test name' and a **description** of either 'dimensional', 'visual' or 'internal'.

**WR3:** Exactly one reference that is the **definiton** in exactly one **action\_property** of type **inspection\_-or\_test\_requirement**.

**WR4:** Exactly one reference that is the **definiton** in exactly one **action\_property** of type **special\_-inspection\_requirement**.

### 5.2.3.1.133 integration\_interval

An **integration\_interval** is a type of **resource\_property**s that is a time step employed for numerical integration, and the total time period covered by a portion of the overall simulation run.

NOTE An **Integration\_interval** is defined according to 4.2.157.

EXPRESS specification:

\*)

```
ENTITY integration_interval
  SUBTYPE OF (action_method);
  WHERE
wr1: (NOT (SIZEOF(QUERY( ama <* USEDIN( SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD' ) |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT (SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' ] *
  TYPEOF(mri)) = 2) AND
  (mri.name = 'delta time'))
  )) = 1 ))) = 0 )))) = 0 )))) = 0 )));
```

```

wr2:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(mri)) = 2)AND
  (mri.name = 'length of elapsed time interval'))
  )) =1 ))) =0 ))))=0 ))))=0));
END_ENTITY; -- integration_interval
(*

```

#### Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** that is referenced by a **property\_definition\_representation** through the **definition** attribute that references through the **used\_representation** attribute a **representation** that references through the **items** attribute a complex instance of **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'delta time'

**WR2:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** that is referenced by a **property\_definition\_representation** through the **definition** attribute that references through the **used\_representation** attribute a **representation** that references through the **items** attribute a complex instance of **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'length of elapsed time interval'.

### 5.2.3.1.134 investment\_casting\_master

An **investment\_casting\_master** is a type of **investment\_casting\_tooling** that is a scale replica of the production tooling needed to produce cast part.

NOTE An **Investment\_casting\_master** is defined according to 4.2.158.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

```
ENTITY investment_casting_master
  SUBTYPE OF (investment_casting_tooling);
  WHERE
```

(\* ----- shrink factor for master tooling ----- \*)

```
WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor for master tooling')) ) =1 )))=1;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'shrink factor for master tooling'.

### 5.2.3.1.135 investment\_casting\_tooling

An **investment\_casting\_tooling** is a type of **product\_definition** that are the reusable metal moulds and components used to form investment patterns and cores to produce precision near\_net shape investment cast parts.

NOTE An **Investment\_casting\_tooling** is defined according to 4.2.159.

EXPRESS specification:

\*)

```
ENTITY investment_casting_tooling
  SUBTYPE OF (casting_product_definition);
```



WHERE

```
(* * to investment_casting_tooling_definition (as investment_tooling) ***
*)
```

```
wr1: ((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description =
  'investment casting tooling definition reference usage') AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_CASTING_TOOLING_DEFINITION'
  IN TYPEOF(sar.relatig_shape_aspect))
  )) >=1 )))) >=1) OR
  (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING'
  IN TYPEOF(pdr.related_product_definition)) AND
  (pdr.name='prototype'))
  ))>=1));
```

```
(* ***** material_definitions ***** *)
```

```
wr2: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='tooling material') AND
  (SIZEOF(QUERY (mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (mfuo.name IN [
  'plaster',
  'expandable polystyrene',
  'iron',
  'soft wood',
  'shell',
  'aluminum',
  'epoxy',
  'wood metal reinforced',
  'brass',
  'hard wood',
  'urethane',
  'other',
  'steel',
  'ceramic',
  'polymethylmethacrylate',
  'plywood'] )
  )) >=1 )) ) =0 ));
```

ISO 10303-223:2008 (E)

```
(* ***** tooling_identifier ***** *)  
  
wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |  
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN  
    TYPEOF(pdr.used_representation)) AND  
    (SIZEOF(QUERY (it <* pdr.used_representation.items |  
      (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'  
        IN TYPEOF(it)) AND (it.name = 'tooling_identifier')))) =1 ))))=1;  
  END_ENTITY; -- investment_casting_tooling  
(*
```

#### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'investment casting tooling definition reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is an **investment\_casting\_tooling\_definition** or there shall be zero or one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** in which the **related\_product\_definition** is a **product\_definition** that is of type **cast\_part\_with\_rigging**.

**WR2:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'tooling material' and of type **make\_from\_usage\_option** with name of 'wood','cast iron','urethane','aluminum' in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'tool identifier'

### 5.2.3.1.136 investment\_casting\_tooling\_definition

An **investment\_casting\_tooling\_definition** is a type of **casting\_product\_definition** that defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces.

NOTE An **Investment\_casting\_tooling\_definition** is defined according to 4.2.160.

EXPRESS specification:

```

*)
ENTITY investment_casting_tooling_definition
  SUBTYPE OF (investment_design_feature);
  WHERE

(* -- to investment_mould (as tree_structure) ----- *)

WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment mould reference usage') AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_MOULD'
  IN TYPEOF(sar.relating_shape_aspect))
  )) <=1 ))) >=1;

(* ----- to sprue_and_runner_mould (as metal_flow_system) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue and runner mould reference usage') AND
  ('CAST_PARTS_SCHEMA.SPRUE_AND_RUNNER_MOULD'
  IN TYPEOF(sar.relating_shape_aspect))
  )) =1 ))) >=1;

(* ----- to production_core_box (as core tooling) ----- *)

WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production core box reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
  IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 ))) >=1;

```

ISO 10303-223:2008 (E)

```

(* ----- to production_investment_cast_mould (as die_mould) -----
*)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling definition occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
      (sar.description =
        'production investment cast mould reference usage') AND
        ( SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD',
          'CAST_PARTS_SCHEMA.LOST_FOAM_CASTING_DIE']
          * TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 )))>=1;

(* ----- to assembly (as pattern_section_assembly) ----- *)

wr5: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.ASSEMBLY'
  IN TYPEOF(adr.assigned_action_method)) ))) =1);

(* ----- to shape_element (as pattern_section_definition)----- *)

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description=
    'investment casting tooling definition occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
      (sar.description = 'pattern section shape usage') AND
      ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
      )) >=1 ))) >=1;

  END_ENTITY; -- investment_casting_tooling_definition
(*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'investment mould reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **investment\_mould**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'sprue and runner mould reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **sprue\_and\_runner\_mould**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production core box reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_core\_box**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production investment cast mould reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_investment\_cast\_mould** or **lost\_foam\_casting\_die**.

**WR5:** Exactly one reference through the **items** attribute of an **applied\_action\_method\_assignment** which is a type of **action\_method\_assignment** that references through the **assigned\_action\_method** an **action\_method** of type **assembly**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'investment casting tooling definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'pattern section shape usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.137 investment\_design\_feature

The **investment\_design\_feature** is a type of **casting\_feature\_definition**, that defines investment features on a cast part.

NOTE A **Investment\_design\_feature** is defined according to 4.2.161.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

```
ENTITY investment_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE

  WR1: SIZEOF(QUERY( pdr <* get_property_definition_representations (SELF) |
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF ( pdr.used_representation ) ) ) =1;

  (* ***** investment_design_feature to orientation (as placement) ***** *)

  WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
    (it.name = 'orientation')))) =1 )))=1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **die\_design\_feature** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **placement** with a **name** of 'orientation'.

### 5.2.3.1.138 investment\_mould

An **investment\_mould** is a type of **casting\_product\_definition** that is the design of a mould constructed of refractory material asround wax or other expendble patterns used for investment casting.

NOTE An **Investment\_mould** is defined according to 4.2.162.

EXPRESS specification:

```

*)
ENTITY investment_mould
  SUBTYPE OF (investment_design_feature);
  WHERE

(* ----- number_of_impressions ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of patterns')) ) =1 ))))=1;

(* ----- mould material ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of patterns')) ) =1 ))))=1;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of patterns'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'mould material'.

**5.2.3.1.139 known\_source**

An **known\_source** is a type of **external\_source** and **pre\_defined\_item**, and it is a source of information whose name and content are pre-determined in the application protocol.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.25 of ISO 10303-224:2006.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)  
ENTITY known_source  
  SUBTYPE OF (external_source,pre_defined_item);  
END_ENTITY;  
(*
```

### 5.2.3.1.140 library\_class\_version\_assignment

An **library\_class\_version\_assignment** is a type of **applied\_external\_identification\_assignment** that specifies a type of external identification.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.26 of ISO 10303-224:2006.

EXPRESS specification:

```
*)  
ENTITY library_class_version_assignment  
  SUBTYPE OF (applied_external_identification_assignment);  
END_ENTITY;  
(*
```

### 5.2.3.1.141 library\_property\_version\_assignment

An **library\_property\_version\_assignment** is a type of **applied\_external\_identification\_assignment** that specifies a type of external identification.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.27 of ISO 10303-224:2006.

EXPRESS specification:

```
*)  
ENTITY library_property_version_assignment  
  SUBTYPE OF (applied_external_identification_assignment);  
END_ENTITY;  
(*
```

### 5.2.3.1.142 loose\_piece

A **loose\_piece** is a type of **casting\_product\_definition** that is required to make one or more features that can not be moulded directly with the production pattern or the required core boxes.

NOTE A **Loose\_piece** is defined according to 4.2.171.



EXPRESS specification:

```

*)
  ENTITY loose_piece
    SUBTYPE OF (sand_cast_design_feature);
  WHERE
wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN ['aluminum','cast iron','urethane','wood'])) ) ) = 1;

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='loose piece shape occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;
END_ENTITY; -- loose_piece
(*

```

Formal propositions:

**WR1:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** of type **make\_from\_usage\_option** in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation** with name of 'aluminum', 'cast iron', 'urethane' or 'wood'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'loose piece occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'loose piece shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.143 lost\_foam\_casting\_die

A **lost\_foam\_casting\_die** is a type of **production\_tool** that that is the tooling to form the pattern, pattern sections, or gating features to make the lost foam pattern.

NOTE A **Lost\_foam\_casting\_die** is defined according to 4.2.170.

EXPRESS specification:

```

*)
  ENTITY lost_foam_casting_die
    SUBTYPE OF (production_tool);
  WHERE

```

ISO 10303-223:2008 (E)

```
(* ***** raw bead size ***** *)

wr1:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'raw bead size')
)))=1))))=0))))=0);

(* ***** steam_line_location ***** *)

wr2:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'steam line location')
)))>=1))))=0))))=0);

(* ***** pattern_removal_instruction ***** *)

wr3:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'pattern removal instruction')
)))>=1))))=0))))=0);

(* -----die_wall_thickness----- *)

wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
(it.name = 'die wall thickness')) ) ) >= 1)) ) = 0)) ) = 0;
```

```

(* ----- to planar_element (as steam_line_location)----- *)

wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
  IN TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name = 'steam line location'))
  )) >= 1)) )) = 1;

(* ----- to planar_element (as fill_gun_location)----- *)

wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
  IN TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name = 'fill gun location')) )) >= 1)) )) = 1;

(* ----- item_size (as fill_gun_size)----- *)

wr7: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
  (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  ((sa_occ.description = 'lost foam casting die occurrence') AND
  (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  ((sar.description = 'fill gun size usage') AND
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)))) |
  (('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sdr.relatng_shape_aspect) )
  )) ) >=1 )) )) )=1 )) ))=0);

(* ----- to cooling port (as mould cooling) ----- *)

WR8: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
  (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  ((sa_occ.description = 'lost foam casting die occurrence') AND
  ( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  ((sar.description = 'cooling port reference usage') AND
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
  (('CAST_PARTS_SCHEMA.COOLING_PORT'
  IN TYPEOF(sdr.relatng_shape_aspect))
  )) ) =1 )) )) )=1 )) ))=0);

```

```
(* ----- to die_clamping (as stationary_die) ----- *)

WR9: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'die clamping reference usage') AND
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
(('CAST_PARTS_SCHEMA.DIE_CLAMPING'
IN TYPEOF(sdr.relating_shape_aspect))
)) ) =1 )) ) ) =1 )) )=0);

(* ----- to machine_mounting (as stationary die mounting) ----- *)

WR10: SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
( 'CAST_PARTS_SCHEMA.MACHINE_MOUNTING'
IN TYPEOF ( pdr.used_representation ) ))) =1;

(* ----- to flask (as defines flask type) ----- *)

WR11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'flask reference usage') AND
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
(('CAST_PARTS_SCHEMA.FLASK' IN TYPEOF(sdr.relating_shape_aspect))
)) ) =1 )) ) ) =1 )) )=0);

(* ----- to gating_design_feature (as gating requirements) ----- *)

WR12: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'gating design feature reference usage') AND
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
(('CAST_PARTS_SCHEMA.GATING_DESIGN_FEATURE'
IN TYPEOF(sdr.relating_shape_aspect))
)) ) >=1 )) ) ) =1 )) )=0);
```

```
(* ----- to slide or die_clamping (as movable die item) ----- *)

WR13: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
  (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  ((sa_occ.description =
  'lost foam casting die occurrence') AND
  ( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (sar.description =
  'slide or die clamping reference usage') AND
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP')
  IN TYPEOF(sar))) ) |
  ( SIZEOF(['CAST_PARTS_SCHEMA.SLIDE',
            'CAST_PARTS_SCHEMA.DIE_CLAMPING']
            * TYPEOF(sdr.relatng_shape_aspect)) = 1)
  )) >=1 )) ) ) >=1 )) )=0);

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr14: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ))) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='tool stored'))=1);
END_ENTITY; -- lost_foam_casting_die
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'raw bead size'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'steam line location'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'pattern removal instruction'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'die wall thickness'.

**WR5:** Exactly one reference that is the **rep\_1** in exactly one **representation\_relationship** in which the **rep\_2** is a **face\_shape\_representation** or **planar\_shape\_representation** with name of 'steam line location'.

**WR6:** Exactly one reference that is the **rep\_1** in exactly one **representation\_relationship** in which the **rep\_2** is a **face\_shape\_representation** or **planar\_shape\_representation** with name of 'fill gun location'.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'fill gun size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'cooling port reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **cooling\_port**.

**WR9:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'die clamping reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **die\_clamping**.

**WR10:** Exactly one **representation** used for the implicit representation shall be of type **machine\_mounting**.

**WR11:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'flask reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **flask**.

**WR12:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'gating design feature reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **gating\_design\_feature**.

**WR13:** Exactly one **shape\_aspect** with a **description** of 'lost foam casting die occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'slide or die clamping reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **slide** or a **die\_clamp**.

**WR14:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the items **attribute** that is of type **person\_and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_role** with **name** of 'tool stored'.

### 5.2.3.1.144 machine

A **machine** is a type of **action\_resource** that defines machines for process planning activities.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.47 of ISO 10303-240:2005.

#### EXPRESS specification:

\*)

```
ENTITY machine
  SUBTYPE OF (action_resource);
  END_ENTITY;
```

(\*

### 5.2.3.1.145 machining\_process

The **machining\_process** is a type of **manufacturing\_process** that defines the type of machine, machine setup, and activity required to perform a automated machining process.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.75 of ISO 10303-240:2005.

#### EXPRESS specification:

\*)

```
ENTITY machining_process
  SUBTYPE OF (manufacturing_process);
  END_ENTITY;
```

(\*

### 5.2.3.1.146 machine\_setup

A **machine\_setup** is an **action\_method** which places or locates the fixture assembly and the part on the machine.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.49 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
  ENTITY machine_setup
    SUBTYPE OF (process_plan_activity);
  END_ENTITY;
(*
```

### 5.2.3.1.147 machine\_allowance

A **machine\_allowance** is a type of **customer\_casting\_requirement** that is additional material dimensions on a cast part to allow for machining.

NOTE A **Machine\_allowance** is defined according to 4.2.175.

EXPRESS specification:

```
*)

ENTITY machine_allowance
  SUBTYPE OF (property_definition);
  WHERE
  wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
    (it.name = 'allowance value')) = 1) )) = 1);

  wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.INSTANCED_FEATURE'
    IN TYPEOF(pdr.related_property_definition.definition)) )) = 1;

  END_ENTITY; -- machine_allowance
(*
```



Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'allowance value'.

**WR2:** There shall be one or more references through the **relating\_property\_definition** by a **property\_definition\_relationship** that references through the **related\_property\_definition** a **property\_definition** that references through the **definition** attribute an **instanced\_feature**.

### 5.2.3.1.148 machine\_element\_relationship

A **machine\_element\_relationship** is an **action\_resource\_relationship** which relates a machine to a machine components.

NOTE This application interpreted model is harmonized with the entity **machining\_process** from paragraph 5.2.3.2.48 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
ENTITY machine_element_relationship
    SUBTYPE OF (action_resource_relationship);
WHERE
    wr1: 'CAST_PARTS_SCHEMA.MACHINE'
        IN TYPEOF(SELF.related_resource);
    wr2: (SIZEOF(['CAST_PARTS_SCHEMA.EJECTOR_EQUIPMENT',
        'CAST_PARTS_SCHEMA.CONTROLLER'] *
        TYPEOF(SELF.relating_resource)) =1) ;
END_ENTITY;
(*
```

Formal proposition:

**WR1:** The **relating\_resource** is a **fixture\_assembly**.

**WR2:** The **related\_resource** is a **controller** or **ejector\_equipment**.

### 5.2.3.1.149 machine\_mounting

A **machine\_mounting** is a type of **representation** that specifies the method used to secure the **Core\_box** on the core machine.

NOTE A **Machine\_mounting** is defined according to 4.2.173.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

ENTITY machine\_mounting

    SUBTYPE OF (representation);

    WHERE

(\* ---- machine\_mounting\_type ---- \*)

```
wr1: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'machine mounting type')
  ))) =1;
```

(\* ---- material\_type ---- \*)

```
wr2: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'material type')
  ))) =1;
```

(\* ---- machine\_mounting to equipment\_size (as dimensions) ---- \*)

```
wr3: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'diameter'))
  ))=1;
```

```
wr4: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'draft'))
  ))=1;
```

```
wr5: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'length'))
  ))=1;
```

```
wr6: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'width'))
  ))=1;
```

```

wr7: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'height'))
  ))=1;

(* ---- machine_mounting to location_element (as location) ---- *)

wr8: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
  IN TYPEOF(ap.rep_2)) ))=1;

(* ---- mount_with ---- *)

wr9: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'mount with') AND
  (it.description IN ['clamps', 'holes', 'slots'])
  )) >=1;

END_ENTITY; -- machine_mounting
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'machine mounting type'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'material type'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'diameter'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'draft'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'length'.

**WR6:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'width'.

**WR7:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'height'.

ISO 10303-223:2008 (E)

**WR8:** Exactly one **representation** used for the implicit representation shall be of type **location\_shape\_representation**.

**WR9:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'mount with' and a **description** of 'clamps','holes' or 'slots'.

### 5.2.3.1.150 machining\_activity

A **machining\_activity** is a type of **manufacturing\_activity** that define any activity that interacts with the part being machined.

NOTE A **Machining\_activity** is defined according to 4.2.174.

EXPRESS specification:

\*)

```
ENTITY machining_activity
  SUBTYPE OF (non_casting_activity);
WHERE

(* ---- to external_schema_definition (as iso_10303_240_data) ----- *)

wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (SIZEOF(QUERY ( edi <* sar.items |
  ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA'
  IN TYPEOF(edi)) ) ) <= 1))))>=0;

  END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the items attribute **externally\_defined\_schema**.

### 5.2.3.1.151 manufacturing\_activity

The **manufacturing\_activity** is a type of **process\_plan\_activity**, that is a type of **action\_method** which is the definition of a process planning **Activity**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.52 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
  ENTITY manufacturing_activity
    SUBTYPE OF (process_plan_activity);
  END_ENTITY;
(*

```

**5.2.3.1.152 manufacturing\_activity\_relationship**

The **manufacturing\_activity\_relationship** is a type of **sequential\_method** that relates a **manufacturing\_process** to either a **manufacturing\_activity**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.53 of ISO 10303-240:2005.

EXPRESS specification:

```

*)

ENTITY manufacturing_activity_relationship
  SUBTYPE OF (sequential_method);
WHERE

  wr1: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
    IN TYPEOF(SELF.relatering_method));

  wr2: ('CAST_PARTS_SCHEMA.MANUFACTURING_ACTIVITY'
    IN TYPEOF(SELF.related_method));

END_ENTITY;
(*

```

Formal propositions:

**WR1:** The **related\_method** attribute shall be **manufacturing\_process**.

**WR2:** The **relatering\_method** attribute shall be **manufacturing\_activity**.

**5.2.3.1.153 manufacturing\_process**

A **manufacturing\_process** is a type of **action\_method** that defines a setup for a specific equipment shall perform manufacturing activities necessary to create a cast part shape.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.87 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
  ENTITY manufacturing_process
    SUBTYPE OF (action_method);
    WHERE

(* - manufacturing_process to casting_activity (as assigned_operation) *)

wr1:  SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
  (('CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP'
  IN TYPEOF(ap)) AND
  ('CAST_PARTS_SCHEMA.CASTING_ACTIVITY'
  IN TYPEOF(ap.related_method))))=1;

(* ----- manufacturing_process to setup_activity (as setup) ----- *)

wr2:(NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
  (('CAST_PARTS_SCHEMA.SINGLE_ACTIVITY_RELATIONSHIP' IN TYPEOF(ap)) AND
  (SIZEOF(['CAST_PARTS_SCHEMA.ANCILLARY_SETUP',
  'CAST_PARTS_SCHEMA.MACHINE_SETUP'] * TYPEOF(ap.related_method)) = 1))
  )=0));

  END_ENTITY;
(*

```

Formal propositions:

**WR1:** There shall be one or more references through the **relating\_method** attribute of an **action\_method\_relationship** of type **single\_setup\_activity\_relationship** that references through the **related\_method** a **casting\_activity**.

**WR2:** There shall be one or more references through the **relating\_method** attribute of an **action\_method\_relationship** of type **single\_setup\_activity\_relationship** that references through the **related\_method** a **ancillary\_setup** or **machine\_setup**.

### 5.2.3.1.154 manufacturing\_process\_relationship

The **manufacturing\_process\_relationship** is a type of **sequential\_method** that relates a **manufacturing\_process** to either a **machining\_process**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.77 of ISO 10303-240:2005.

EXPRESS specification:

```

*)

ENTITY manufacturing_process_relationship
    SUBTYPE OF (sequential_method);

WHERE

    wr1: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
        IN TYPEOF(SELF.related_method));

    wr2: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
        IN TYPEOF(SELF.relying_method));

END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one reference that is the **related\_method** in exactly one **action\_method\_relationship** in which the **related\_method** is a **manufacturing\_process**.

**WR2:** Exactly one reference that is the **related\_method** in exactly one **action\_method\_relationship** in which the **relying\_method** is a **manufacturing\_process**.

**5.2.3.1.155 master\_pattern**

A **master\_pattern** is a type of **sand\_casting\_tooling** that is a scaled replica used for making a casting that will become the production pattern needed to produce the cast part.

NOTE A **Master\_pattern** is defined according to 4.2.181.

EXPRESS specification:

```

*)

ENTITY master_pattern
    SUBTYPE OF (sand_casting_tooling);
    WHERE

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        ('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) )) = 1;

```

## ISO 10303-223:2008 (E)

```
(* ----- shrink factor for master pattern ----- *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'shrink factor for master pattern')
    ))) =1 ))))=1;

(* ** to machining_allowance (as master_allowance_on_master_pattern) * *)

wr3: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION' ) |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))=1);
END_ENTITY; -- master_pattern
(*
```

### Formal propositions:

**WR1:** The **casting\_feature\_definition** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'shrink factor for master pattern'.

**WR3:** Exactly one reference by a **property\_definition** through the **definition** attribute and the **property\_definition** is of type **machine\_allowance**.

### 5.2.3.1.156 master\_sample

A **master\_sample** is a type of **cast\_part** that is an inspected part that has been produced by a casting process and is identified by the supplier as the master sample.

NOTE A **Master\_sample** is defined according to 4.2.182.

### EXPRESS specification:

\*)

```
ENTITY master_sample
  SUBTYPE OF (cast_part);
WHERE
```

```
(* ----- master_sample to date (as production_date) *)

wr1: SIZEOF(QUERY (da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS' ) |
  ((da.role.name='production date') AND
  ('CAST_PARTS_SCHEMA.CALENDAR_DATE'
  IN TYPEOF(da\date_assignment.assigned_date))))=1;
```



```

(* --- master_sample to design_part (as product_specification_used) -- *)

wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  ('CAST_PARTS_SCHEMA.DESIGN_PART'
  IN TYPEOF(pdr.relatng_product_definition)) ))=1;

(* ----- master_sample to heat (as belongs_to_heat) -- *)

wr3:  SIZEOF(QUERY(da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS' ) |
  ('CAST_PARTS_SCHEMA.DATED_EFFECTIVITY'
  IN TYPEOF(da\effectivity_assignment.assigned_effectivity)
  ))=1;

(* ----- master_sample to lot (as belongs_to_lot) -- *)

wr4:  SIZEOF(QUERY(da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS' ) |
  ('CAST_PARTS_SCHEMA.LOT_EFFECTIVITY'
  IN TYPEOF(da\effectivity_assignment.assigned_effectivity)
  ))=1;

(* ----- master_sample to process_plan (as process_plan_used) ----*)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.DEFINED_PRODUCT') |
  ((pdr.description='process plan used') AND
  ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(pdr.process)) )
  )<=1;

WR6: SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.EXTERNAL_SOURCE'
  IN TYPEOF(pdr\external_identification_assignment.source)) AND
  (NOT (SIZEOF(QUERY(es <* USEDIN(pdr.source,
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM.SOURCE') |
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA' IN TYPEOF(es)
  )) <=1 )) ) ) )=0;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** There shall be exactly one reference from an **applied\_date\_assignment** through the **items** attribute that is of type **date\_assignment** that references through the **role** attribute a **date\_role** with **name** of 'production date' and references through the **assigned\_data** a **calendar\_date**.

**WR2:** There shall be exactly one reference from an **product\_definition\_relationship** through the **related\_product\_definition** attribute that references a **design\_part** through the **relating\_product\_definition**.

**WR3:** There shall be exactly one reference from an **applied\_effectivity\_assignment** through the **items** attribute that is of type **effectivity\_assignment** that references through the **assigned\_effectivity** attribute an **effectivity** of type **dated\_effectivity**.

**WR4:** There shall be exactly one reference from an **applied\_effectivity\_assignment** through the **items** attribute that is of type **effectivity\_assignment** that references through the **assigned\_effectivity** attribute an **effectivity** of type **lot\_effectivity**.

**WR5:** There shall be exactly one reference from an **process\_product\_association** through the **defined\_product** with **name** 'process plan used' and references through the **process** attribute a **product\_definition\_process** of type **process\_plan\_version**.

**WR6:** There shall be zero or one reference from an **applied\_external\_identification\_assignment** through the **items** attribute that is of type **external\_identification\_assignment** that references through the **source** attribute an **external\_source** that is referenced by an **externally\_defined\_item** through the **source** attribute of type **externally\_defined\_schema**.

### 5.2.3.1.157 material\_structure

A **material\_structure** is a type of **material\_property** that is the representation of the physical characteristics of a material.

NOTE A **Material\_structure** is defined according to 4.2.188.

EXPRESS specification:

\*)

```
ENTITY material_structure
    SUBTYPE OF (material_property);
WHERE
```

```
(* --- material_structure to material_structure_element (as element)-- *)
```

```
WR1: (SELF.definition.frame_of_reference.name=
    'material structure definition');
```

```
(* --- material_structure to illustration (as illustration)----- *)

wr2: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
  (adr.assigned_document))))=1;

(* --- material_structure to specification (as referenced_to)----- *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
  (adr.assigned_document))))=1;
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one reference that is the **definition** in exactly one **property\_definition\_representation** of type **material\_property\_representation** with name of 'material structure element'.

**WR2:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **illustration**.

**WR3:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is a **document**.

### 5.2.3.1.158 material\_usage

A **material\_usage** is a type of **material\_property** that is the kind and amount of material used in a **process\_plan\_activity**.

NOTE A **Material\_usage** is defined according to 4.2.190.

#### EXPRESS specification:

```
*)

ENTITY material_usage
  SUBTYPE OF (property_definition);
  WHERE

(* ---- Material_usage to numeric_parameter (as amount_used) ----- *)

wr1:(NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ( (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))= 2) AND
  (it.name = 'amount used')) ) =1 )) ) =0));
```

ISO 10303-223:2008 (E)

```
(* ----- material_usage to material (as material_used) -----*)
wr2: ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (SELF.definition));
    END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'amount used'.

**WR2:** Exactly one reference that is the **definition** in exactly one **product\_definition** that is the product definition for material.

### 5.2.3.1.159 material\_usage\_record

A **material\_usage\_record** is a type of **activity\_result\_record** that is a log of material used during a **process\_plan\_activity**.

NOTE A **Material\_usage\_record** is defined according to 4.2.191.

EXPRESS specification:

```
*)
ENTITY material_usage_record
  SUBTYPE OF (activity_result_record);
  WHERE

  (* ----- Material_usage to numeric_parameter (as amount_used) ----- *)
  wr1: (SIZEOF(QUERY ( it <* SELF.items |
    ( (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
      (it.name = 'actual substance usage amount')) ) = 1);

  (* ----- material_usage_record to material_usage (as recorded_usage)----- *)
  wr2: (SIZEOF(QUERY( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('CAST_PARTS_SCHEMA.MATERIAL_USAGE' IN TYPEOF(pdr.definition))))=1);

  END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'actual substance usage amount'.

**WR2:** There shall be exactly one reference through the **used\_representaion** attribute by **property\_-definiton\_representation** that is of type **material\_usage**.

### 5.2.3.1.160 mechanical\_property\_requirements

A **mechanical\_property\_requirements** is a type of **activity\_result\_record** that are those property requirements of a material that reveal the elastic and inelastic properties when force is applied.

NOTE A **Mechanical\_property\_requirements** is defined according to 4.2.192.

EXPRESS specification:

\*)

```
ENTITY mechanical_property_requirements
  SUBTYPE OF (property_definition);
  WHERE
```

```
(* ----- chemical_analysis_certification ----- *)
```

```
wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'chemical analysis certification') AND
  (it.description IN (['TRUE','FALSE']))
  ) )) =1 )))=0));
```

```
(* ----- mechanical_properties_certification ----- *)
```

```
wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'mechanical properties certification') AND
  (it.description IN (['TRUE','FALSE']))
  ) )) =1 )))=0));
```

ISO 10303-223:2008 (E)

```
(* ----- tensile_strength ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'tensile strength')
  ))) =1 )))=0));

(* ----- yeild_strength ----- *)

wr4: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'yeild strength')
  ))) =1 )))=0));

(* ----- to material_hardness (as hardness)----- *)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.MATERIAL_PROPERTY'
  IN TYPEOF(pdr.related_property_definition)) )) = 1;

(* -----to specification (as requirement_specification) ----- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'requirement specification') ))
= 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'chemical analysis certification' and a **description** of either 'true' or 'false'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'mechanical properties certification' and a **description** of either 'true' or 'false'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'tensile strength'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'yeild strength'.

**WR5:** Exactly one reference of **definition** is a **action** of type **property\_process** that has exactly one reference that is the **process** in exactly one **process\_property\_association** that has exactly one reference of **property\_or\_shape** in exactly one **property\_definition** of type **paterial\_property**.

**WR6:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** in which the **kind** is exactly one **document\_kind** with **product\_data\_type** of 'requirement specification'.

### 5.2.3.1.161 mechanical\_stress

A **mechanical\_stress** is a type of **action\_property** that is the type of design load with the maximum load value an in-service application will apply to the cast part.

NOTE A **Mechanical\_stress** is defined according to 4.2.193.

#### EXPRESS specification:

\*)

```
ENTITY mechanical_stress
    SUBTYPE OF (customer_casting_requirement);
WHERE
```

(\* ----- to descriptive\_parameter(as type\_of\_stress) ----- \*)

```
wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'type of stress')
    ) ) = 1 )))=0));
```

(\* ----- to numeric\_parameter(as maximum\_psi\_design\_stress) ----- \*)

```
wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'maximum psi design stress')
    ) ) = 1 )))=0));
```

ISO 10303-223:2008 (E)

```
(* ----- to numeric_parameter(as percent_safety_factor_required) ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'percent safety factor required'))
)) = 1 )))=0));
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'type of stress'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'maximum psi design stress'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'percent safety factor required'.

### 5.2.3.1.162 Melting\_equipment

A **melting\_equipment** is a type of **casting\_equipment** that produces enough heat to change a solid material to a liquefied state.

NOTE A **Melting\_equipment** is defined according to 4.2.194.

#### EXPRESS specification:

```
*)
ENTITY melting_equipment
  SUBTYPE OF (casting_equipment);
END_ENTITY;
(*
```

### 5.2.3.1.163 metalcaster\_simulation

A **metalcaster\_simulation** is a type of **simulation\_process** that is a request from a foundry to use a simulation software tool to produce a simulation of a casting process for a proposed cast part.

NOTE A **Metal\_caster\_simulation** is defined according to 4.2.196.



EXPRESS specification:

```

*)

ENTITY metalcaster_simulation
  SUBTYPE OF (simulation_process);
  WHERE

(* ----- to simulation_exception_report (as defect_results) -----*)

wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc <* SELF.documents |
  (doc.kind.product_data_type='simulation exception report')))=1);

(* ----- to tooling_report (as suggestion_for_changes) -----*)

wr2: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc <* SELF.documents |
  (doc.kind.product_data_type='simulation tooling report')))=1);

(* ---- to gating_system (as proposed_rigging) ----- *)

wr3:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ---- to pattern_equipment_available (as available_tooling) ----- *)

wr4:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PATTERN_EQUIPMENT_AVAILABLE' IN TYPEOF (eds))
  ))) <=1 ))))=0));
END_ENTITY;
(*

```

Formal propositions:

**WR1: Metalcaster\_simulation** shall be of type **action\_method\_with\_associated\_documents** and shall be zero or more references to a **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'simulation exception report'.

**WR2: Metalcaster\_simulation** shall be of type **action\_method\_with\_associated\_documents** and shall be zero or more references to a **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'simulation tooling report'.

**WR3:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment**, the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is exactly one **property\_definition** that references a **gating\_system** through the **definition** attribute.

**WR4:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** of type **pattern\_equipment\_available**.

### 5.2.3.1.164 meshing\_condition

A **meshing\_condition** is a type of **resource\_property** that is a description of the fineness of mesh needed in a region.

NOTE A **Meshing\_condition** is defined according to 4.2.195.

EXPRESS specification:

```

*)
  ENTITY meshing_condition
    SUBTYPE OF (action_method);
  WHERE

    (* number_of_elements *)

wr1:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(mri\measure_with_unit.value_component))AND
  (mri.name = 'number of elements'))
  )) =1 ))) =0 )))))=0 )))))=0));

  (* number_of_nodes *)

wr2:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(mri\measure_with_unit.value_component))AND
  (mri.name = 'number of nodes'))
  )) =1 ))) =0 )))))=0 )))))=0));

  (* meshing_condition to numeric_range(as density_of_nodes) *)

wr3:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(mri)) = 2)AND
  (mri.name = 'density of nodes'))

```

ISO 10303-223:2008 (E)

```
)) =1 ))) =0 ))))=0 ))))=0));

(* meshing_condition to numeric_range(as density_of_elements) *)

wr4:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ((SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(mri)) = 2)AND
  (mri.name = 'density of elements'))
  )) =1 ))) =0 ))))=0 ))))=0));

(* meshing_condition to shape_aspect (as area_to_be_meshed) *)

wr5:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
  )) =1 )) ))=0));

END_ENTITY; -- meshing_condition
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of elements'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of nodes'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** with a **name** of 'density of nodes'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** with a **name** of 'density of elements'.

**WR5:** Exactly one reference of **resource** is an **action\_resource\_requirement** that has exactly one reference that is the **usage** is an **action** of type **property\_process** that has exactly one reference that is the **process** in exactly one **process\_property\_association** that has exactly one reference of **property\_or\_shape** in exactly one **shape\_aspect**.

### 5.2.3.1.165 mismatch\_tolerance

A **metal\_casters\_casting\_requirement** is a type of **customer\_casting\_requirement** that is an imperfection on a cast part caused by inaccurate alignment or displacement or dimensional differences between the mould or die components, producing a step on the surface along the parting line.

NOTE A **Mismatch\_tolerance** is defined according to 4.2.197.

#### EXPRESS specification:

```

*)
ENTITY mismatch_tolerance
    SUBTYPE OF (property_definition);
WHERE

(* ----- to numeric_parameter(as maximum_amount) ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component))) AND
    (it.name = 'maximum amount')
    )) =1 )) ) = 0);

(* ----- to parting_surface(as with_respect_to)----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (NOT(SIZEOF(QUERY (ps <* USEDIN(pdr.related_property_definition,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'CAST_PARTS_SCHEMA.PARTING_SURFACE' IN TYPEOF(ps)))=1))))=0);

END_ENTITY;
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'maximum amount'.

**WR2:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface**.

### 5.2.3.1.166 mould\_box

A **mould\_box** is a type of **casting\_product\_definition** that is designed to mould a cast part that requires no metal jacket around the mould when the mould box is removed from the mould.

NOTE A **Mould\_box** is defined according to 4.2.199.

EXPRESS specification:

```

*)
ENTITY mould_box
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- mould_box to item_size (as box_size) ----- *)

WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='mould box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'mould box dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

(* ***** mould_box to descriptive_parameter (as description) ***** *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'mould box description')))) <=1 ))))=1;
END_ENTITY; -- mould_box
(*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'mould box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'mould box dimension usage' in which the **relying\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'mould lock description'.

### 5.2.3.1.167 mould\_lock

A **mould\_lock** is a type of **casting\_feature\_definition** that are forms located, inside the flask line, on the pattern parting board or plate that makes an impression in the mould sand that will align the cope and drag moulds, when the moulds are closed, to insure a match of the cast part at the parting line.

NOTE A **Mould\_lock** is defined according to 0.

EXPRESS specification:

```

*)

ENTITY mould_lock
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- mould_locks to item_size (as lock_dimensions) ----- *)

WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='mould lock occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
  (sar.description = 'mould lock dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

(* ***** mould_locks to descriptive_parameter (as description) ***** *)

wr2:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'mould lock description')))) <=1
  )))=1;
END_ENTITY; -- mould_lock
(*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'mould lock occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'mould lock usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'mould lock description'.

### 5.2.3.1.168 moulding\_equipment

A **moulding\_equipment** is a type of **equipment** that is the machinery used to produce sand moulds from production patterns.

NOTE A **Moulding\_equipment** is defined according to 4.2.201.

EXPRESS specification:

```
*)
ENTITY moulding_equipment
  SUBTYPE OF (casting_equipment);
END_ENTITY;
(*
```

**5.2.3.1.169 non\_casting\_activity**

A **non\_casting\_activity** is a type of **casting\_activity** that is a procedure or process that is required to be completed on the cast part after the casting has been poured.

NOTE A **Non\_casting\_operation** is defined according to 4.2.202 and defines the requirements for **non\_casting\_activity**.

EXPRESS specification:

```
*)
ENTITY non_casting_activity
  SUBTYPE OF (casting_activity);
  WHERE

(* ----- to descriptive_parameter (as operation description) ----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'operation description')
  ))) =1 ))))>=1;

(* ---- to finishing_or_machining_operations (as requirement) ----- *)

wr2: SIZEOF(QUERY ( sar <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (SIZEOF(QUERY ( edi <* sar.items |
  ('CAST_PARTS_SCHEMA.FINISHING_AND_MACHINING_OPERATION'
  IN TYPEOF(edi)) )) <= 1))))>=0;
```



```
(* --- to cast_part (as part_geometry_with_machining_allowance) ----- *)
(* --- to design_part (as part_geometry_with_no_machining_allowance) -- *)

wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (NOT(SIZEOF(QUERY ( ip <* sar.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.CAST_PART',
  'CAST_PARTS_SCHEMA.DESIGN_PART'] * TYPEOF(ip)) <=1))
  ) =0 ))))>=0);
END_ENTITY;
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'operation description'

**WR2:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** of type **finishing\_and\_machining\_operation**.

**WR3:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **product\_definition** of type **cast\_part** or **design\_part**.

### 5.2.3.1.170 non\_casting\_equipment\_process

The **non\_casting\_equipment\_process** is a type of **non\_machining\_process** that defines **casting\_activities** for non equipment processes.

NOTE This application interpreted model is harmonized with the **non\_machining\_process** object from paragraph 5.2.3.1.69 of ISO 10303-240:2005.

#### EXPRESS specification:

```
*)
  ENTITY non_casting_equipment_process
    SUBTYPE OF (non_machining_process);
  END_ENTITY;
(*
```

### 5.2.3.1.171 non\_machining\_process

The **non\_machining\_process** is a type of **manufacturing\_process** that defines **manufacturing\_activities** for non machining processes.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.58 of ISO 10303-240:2005.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

```
ENTITY non_machining_process
  SUBTYPE OF (manufacturing_process);
END_ENTITY; -- non_equipment_process
(*
```

### 5.2.3.1.172 non\_permanent\_moulding\_process

A **non\_permanent\_moulding\_process** is a type of **casting\_activity** that is a process where the mould is destroyed when the metal casting has been removed from the mould.

NOTE A **Non\_permanent\_moulding\_process** is defined according to 4.2.204.

EXPRESS specification:

\*)

```
ENTITY non_permanent_moulding_process
  SUBTYPE OF (casting_activity);
WHERE
```

```
(* ----- moulding_process ----- *)
```

```
wr1: (SELF.description IN ['shell','green sand','no bake',
                          'lost foam','vacuum']);
```

```
(* ----- number_of_impressions ----- *)
```

```
wr2:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
   TYPEOF(pdr.representation)) AND
   (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of impressions')) ) =1 ))))=1;
```

```

(* ----- pattern_form ----- *)

wr3:      SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'pattern form') AND
  (it.description IN [
'mounted match plate',
'mounted cope and drag inserts',
'mounted cope and drag boards',
'mounted flaskless cope and drag',
'loose split pattern',
'loose pattern with follow block',
'sweep'])))) =1 ))))>=1;

(* ---- non_permanent_moulding_process to flask (as flask_type) ----- *)
(* ---- non_permanent_moulding_process to flaskless (as flask_type) ---- *)
(* ---- non_permanent_moulding_process to mould_box (as flask_type) ---- *)

wr4:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  (SIZEOF(['CAST_PARTS_SCHEMA.FLASK',
'CAST_PARTS_SCHEMA.FLASKLESS'] * TYPEOF(eds.definition)) =1)
  ))) =1 )))) =0));

(* ---- to production_pattern_definition (as pattern) ----- *)

wr5:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  (SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION',
'CAST_PARTS_SCHEMA.LOST_FOAM_CASTING_DIE',
'CAST_PARTS_SCHEMA.INVESTMENT_MOULD'] * TYPEOF(eds.definition))
=1)
  ))) =1 )))) =0));

  END_ENTITY; -- non_permanent_moulding_process
(*

```

Formal propositions:

**WR1:** There shall be exactly one **description** of either 'shell', 'no bake', 'lost form', 'gas cured' or 'green sand'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of impressions'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'pattern form' and a **description** of either 'mounted match plate', 'mounted cope and drag inserts', 'mounted cope and drag boards', 'mounted flaskless cope and drag', 'loose split pattern', 'loose pattern with follow block' or 'sweep'

**WR4:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** in which the **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **property\_definition** that has the **definition** attribute reference a **characterized\_object** of type **flask** or **flaskless**.

**WR5:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** in which the **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **property\_definition** that has the **definition** attribute reference a **characterized\_object** of type **production\_pattern\_definition**, **lost\_foam\_casting\_die** or **investment\_mould**.

### 5.2.3.1.173 ordered\_part

An **ordered\_part** represents the measured amount of product that is to be manufactured.

NOTE 1 An **Ordered\_part** is defined according to 4.2.209.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.29 of ISO 10303-224:2006.

EXPRESS specification:

\*)

```
ENTITY ordered_part
  SUBTYPE OF (action_assignment, characterized_object);
  items : SET [1:?] OF ordered_item;
  WHERE
wr1: (SIZEOF(USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION')) = 1);

wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) )
  = 0);
```

```

wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF( pdr.used_representation.items) = 1)) )) = 0)) )) = 0);

wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'quantity')) )) = 1)) ))
= 0)) )) = 0);
END_ENTITY; -- ordered_part
(*

```

Attribute definition:

**items:** the set of **product\_definition\_formations** for which an **action** is being assigned.

Formal propositions:

**WR1:** The **ordered\_part** shall have exactly one property defined for it.

**WR2:** The property defined for the **ordered\_part** shall have exactly one representation.

**WR3:** The representation of the property defined for the **ordered\_part** shall contain exactly one **representation\_item** in its set of **items**.

**WR4:** The **representation\_item** in the set of **items** for the representation of the property for the **ordered\_part** shall be of type **measure\_representation\_item** with a **name** of 'quantity'.

### 5.2.3.1.174 other\_inspection

An **other\_inspection** is a type of **inspection\_or\_test\_result** that is the outcome of an examination or measurement and comparing the resultant to the design requirement.

NOTE An **Other\_inspection** is defined according to 4.2.212.

EXPRESS specification:

```

*)
  ENTITY other_inspection
    SUBTYPE OF (inspection_or_test_result);
    WHERE

  (* ----- inspection result ----- *)

  wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'inspection result')
    ) ))=1 )))) = 0);

  (* -- to inspection_or_test_requirement (as requirements)----- *)

  wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_REQUIREMENT'
    IN TYPEOF(pdr.related_property_definition)
    )) =1);
  END_ENTITY;

  (*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'inspection result'.

**WR1:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_definition\_relationship** in which the **relating\_property\_definition** is an **inspection\_or\_test\_requirement**.

### 5.2.3.1.175 parting\_surface

A **parting\_surface** is a type of **shape\_representation** that represents geometry defined by a portion of an associated surface for the use as a parting surface as defined as the application object.

NOTE A **Parting\_surface** is defined according to 4.2.218.

EXPRESS specification:

```

*)
ENTITY parting_surface
  SUBTYPE OF (shape_representation);
  WHERE

  wr1: (SIZEOF(QUERY( it <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'cope or drag') AND
    (it.description IN ['cope','drag']) )))=1);

  wr2: (SIZEOF(QUERY( it <* SELF.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'male or female') AND
    (it.description IN ['male','female']) )))=1);

  wr3: (SIZEOF(QUERY( it <* SELF.items |
    (SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'offset distance'))>=1);

  wr4: (SIZEOF(QUERY (rep <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
    (SIZEOF(['CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' ,
    'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION']
    * TYPEOF(rep.rep_2))>=1)))=1);
END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'cope or drag' and a **description** of either 'cope' or 'drag'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'male or female' and a **description** of either 'male' or 'female'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'offset distance'.

**WR4:** Exactly one reference that is the **rep\_1** in exactly one **representation\_relationship** in which the **rep\_2** is a **face\_shape\_representation** or **planar\_shape\_representation**.

### 5.2.3.1.176 pattern\_equipment\_available

A **pattern\_equipment\_available** is a type of **customer\_casting\_requirement** that declares the customer has pattern tooling and the tooling configuration.

NOTE A **Pattern\_equipment\_available** is defined according to 4.2.220.

EXPRESS specification:

```

*)
ENTITY pattern_equipment_available
    SUBTYPE OF (customer_casting_requirement);
WHERE

(* ----- core_box_cavity_id ----- *)

wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'core box cavity id')
    ) ) =1 )))=0));

(* ----- core_box_id ----- *)

wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'core box id')
    ) ) =1 ))) =0));

(* ----- number_of_core_boxes ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component))) AND
    (it.name = 'number of core boxes')
    ) ) =1 )))=0));
    
```



```

(* ----- number_of_cores_per_box ----- *)

wr4:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of core per box')
  )) =1 ))) =0));

(* ----- number_of_cores_per_casting ----- *)

wr5:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of cores per casting')
  )) =1 ))) =0));

(* ----- number_of_patterns ----- *)

wr6:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of patterns')
  )) =1 ))) =0));

(* ----- condition ----- *)

wr7:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'condition') AND
  (it.description IN ['good','bad','fair','new'] )
  ))) =1 )))=0));

```

ISO 10303-223:2008 (E)

```
(* ----- core_box_material ----- *)

wr8: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
((pdr.name = 'core box material') AND
(SIZEOF(QUERY ( mfuo <*
USEDIN(pdr.related_property_definition.definition,
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
(mfuo.name IN [
'plaster',
'expandable polystyrene',
'iron',
'soft wood',
'shell',
'aluminum',
'epoxy',
'wood metal reinforced',
'brass',
'hard_wood',
'urethane',
'other',
'steel',
'ceramic',
'polymethylmethacrylate',
'plywood']) ))
= 1)) )) =0));

(* ----- pattern_impression_id ----- *)

wr9: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'pattern impression id')
))) =1 )))=0));
```

```
(* ----- pattern_material ----- *)

wr10: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATINGS_PROPERTY_DEFINITION') |
  ((pdr.name = 'core box material') AND
  (SIZEOF(QUERY ( mfuo <*
  USEDIN(pdr.related_property_definition.definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (mfuo.name IN [
  'plaster',
  'expandable polystyrene',
  'iron',
  'soft wood',
  'shell',
  'aluminum',
  'epoxy',
  'wood metal reinforced',
  'brass',
  'hard wood',
  'urethane',
  'other',
  'steel',
  'ceramic',
  'polymethylmethacrylate',
  'plywood']) ))
  = 1)) )) =0));

(* ----- core_per_core_box_per_assembly ----- *)

wr11: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'core per core box per assembly') AND
  (it.description IN ['same core','different core'] )
  ))) =1 )))=0));
```

ISO 10303-223:2008 (E)

```
(* ----- pattern ----- *)

wr12: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'pattern') AND
  (it.description IN [ 'loose pattern with follow on board',
    'loose solid pattern',
    'loose split pattern', 'mounted cope and drag boards',
    'mounted cope and drag flaskless', 'mounted cope and drag inserts',
    'mounted snapboard', 'mounted match plate', 'sweep pattern'] )
  ))) =1 )))=0));

(* ----- to flask (as flask_size_for_mounted_patterns)----- *)

wr13: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flask size for mounted patterns') AND
  ('CAST_PARTS_SCHEMA.FLASK'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);

(* ----- to flask (as flask_size_for_plate_patterns)----- *)

wr14: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flask size for plate pattern') AND
  ('CAST_PARTS_SCHEMA.FLASK'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);

(* ----- to flaskless(as flaskless_pattern)----- *)

wr15: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flaskless pattern') AND
  ('CAST_PARTS_SCHEMA.FLASKLESS'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);
```

```

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr16: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ('CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF(pdr.used_representation)
  ))) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='tooling_location')))=1);

END_ENTITY; -- pattern_equipment_available
(*

```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core box cavity id'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core box id'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of core boxes'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of cores per box'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of cores per casting'.

**WR6:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number patterns'.

**WR7:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'condition' and a **description** of either 'good', 'bad', 'fair' or 'new'.

**WR8:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core box material' and a **description** of either 'plaster', 'expandable polystyrene', 'iron', 'soft wood', 'shell', 'aluminum', 'epoxy', 'wood metal reinforced', 'brass', 'hard\_wood', 'urethane', 'other', 'steel', 'ceramic', 'polymethylmethacrylate' or 'plywood'.

**WR9:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'pattern impression id'.

**WR10:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'pattern material' and a **description** of either 'plaster', 'expandable polystyrene', 'iron', 'soft wood', 'shell', 'aluminum', 'epoxy', 'wood metal reinforced', 'brass', 'hard\_wood', 'urethane', 'other', 'steel', 'ceramic', 'polymethylmethacrylate' or 'plywood'.

**WR11:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core per core box per assembly' and a **description** of either 'same core' or 'different core'.

**WR12:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'pattern' and a **description** of either 'loose pattern with follow on board', 'loose solid pattern', 'loose split pattern', 'mounted cope and drag boards', 'mounted cope and drag flaskless', 'mounted cope and drag inserts', 'mounted snapboard', 'mounted match plate', 'sweep pattern'.

**WR13:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** with name of 'flask size for mounted patterns' in which the **related\_property\_definition** is a **property\_definition** that references through the **definition** attribute a **characterized\_object** of type **flask**.

**WR14:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** with name of 'flask size for plate pattern' in which the **related\_property\_definition** is a **property\_definition** that references through the **definition** attribute a **characterized\_object** of type **flask**.

**WR15:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** with name of 'flaskless pattern' in which the **related\_property\_definition** is a **property\_definition** that references through the **definition** attribute a **characterized\_object** of type **flaskless**.

**WR16:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the **items** attribute that is of type **person\_and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_role** with **name** of 'tooling location'.

### 5.2.3.1.177 pattern\_plate

A **pattern\_plate** is a type of **casting\_product\_definition** that is a metal plate or wooden board that one or more patterns and a gating system are mounted to, that establishes a parting surface for the cope and for the drag sections of a mould.

NOTE A **Pattern\_plate** is defined according to 4.2.221.

EXPRESS specification:

```

*)
ENTITY pattern_plate
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- type_of_plate ----- *)

wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN ['wood', 'metal', 'metal insert', 'wood insert'])) )
  = 1;

(* ----- pattern_plate to item_size (as pattern_plate_dimensions) --- *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='pattern_plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pattern_plate usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) =1 ))) >=1;

(* - pattern_plate to location_element (as mounting_hole_locations)---- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='location')))=1;

(* ----- pattern_plate to flask (as defines_flask_type) ----- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='pattern_plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'flask reference usage') AND
  ('CAST_PARTS_SCHEMA.FLASK' IN TYPEOF(sar.relateing_shape_aspect))
  )) =1 ))) >=1;

```

ISO 10303-223:2008 (E)

(\* ----- pattern\_plate to gating\_system (as gate\_rigging) ----- \*)

```
wr5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='pattern plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gating system reference usage') AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM'
  IN TYPEOF(sar.relater_shape_aspect))
  )) =1 ))) >=1;
```

(\* ----- pattern\_plate to mould\_locks (as lock\_definitions) ----- \*)

```
wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='pattern plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'mould lock reference usage') AND
  ('CAST_PARTS_SCHEMA.MOULD_LOCK' IN TYPEOF(sar.relater_shape_aspect))
  )) <=1 ))) >=1;
```

(\* ----- pattern\_plate to parting\_surface (as parting\_line) ---- \*)

```
wr7: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.PARTING_SURFACE' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='parting line')))=1;
```

(\* ----- pattern\_plate to pin\_center (as pin\_center\_definition) -- \*)

```
wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='pattern plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pin center reference usage') AND
  ('CAST_PARTS_SCHEMA.PIN_CENTER' IN TYPEOF(sar.relater_shape_aspect))
  )) <=1 ))) >=1;
```



```
(* ----- pattern_plate to moulding_equipment (as moulding_machine)----- *)

wr9: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
(('CAST_PARTS_SCHEMA.ACTION_METHOD'
IN TYPEOF(adr.assigned_action_method)) AND
(SIZEOF(QUERY(ar <* USEDIN(adr.assigned_action_method,
'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
('CAST_PARTS_SCHEMA.MOULDING_EQUIPMENT' IN TYPEOF(ar) )))=1))))=1);

END_ENTITY; -- pattern_plate
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **definition** in exactly one **material\_designation**, with **name** of 'wood', 'metal insert', 'metal' or 'wood insert'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'pattern plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'pattern plate usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **location\_shape\_representation** through the **used\_representation** that has a **name** of 'location'.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'pattern plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'flask reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **flask**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'pattern plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'gating system reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **gating\_system**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'pattern plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'mould lock reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **mould\_lock**.

**WR7:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'parting line'.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'pattern plate occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'pin center reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **pin\_center**.

**WR9:** Exactly one reference that is the **items** in exactly one **applied\_action\_method\_assignment** of type **action\_method\_assignment** which the **assigned\_action\_method** was an **action\_method** that has one reference that is the **usage** in exactly one **action\_resource** of type **moulding\_equipment**.

### 5.2.3.1.178 payment\_and\_shipping

A **payment\_and\_shipping** is a type of **customer\_casting\_requirement** that defines the method of and requirements for payment. **Payment\_and\_shipping** also designates the method of shipping the cast parts to the customer.

NOTE A **Payment\_and\_shipping** is defined according to 4.2.223.

EXPRESS specification:

```

*)
ENTITY payment_and_shipping
    SUBTYPE OF (customer_casting_requirement);
WHERE

(* ----- customer_payment ----- *)

wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'customer payment')
    ) )) =1 ))=0));

(* ----- packaging_or_creating ----- *)

wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'packaging or creating')
    ) )) =1 ))=0));
    
```

```
(* ----- ship_casting ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'ship_casting')
) )) =1 )))=0));
END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'customer payment'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'packaging or creating'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'ship casting'.

### 5.2.3.1.179 permanent\_moulding\_process

A **permanent\_moulding\_process** is a type of **casting\_activity** that is a process where the mould is used repeatedly many times to produce metal castings and for high production requirements.

NOTE A **Permanent\_moulding\_process** is defined according to 4.2.226.

#### EXPRESS specification:

\*)

```
ENTITY permanent_moulding_process
  SUBTYPE OF (casting_activity);
  WHERE
```

```
(* ----- moulding_process ----- *)
```

```
wr1: (SELF.description IN ['static','tilt','low pressure']);
```

ISO 10303-223:2008 (E)

```
(* ----- number_of_impressions ----- *)

wr2:      SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of impressions')) ) ) =1 ))))=1;

(* ----   to die_mould_vent (as ejection_definition) ----- *)

wr3: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.DIE_MOULD_VENT' IN  TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ----   to ejector_system (as vent_definition) ----- *)

wr4: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM' IN  TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ----   (as permanent_die_definition) ----- *)

wr5: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD' IN  TYPEOF(eds.definition))
  ))) =1 ))))=0));
```

```

(* -- permanent_mould_process to ingate (as metal_flow_system) ----- *)

wr6:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.INGATE' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ----- p to runner (as metal_flow_system) ----- *)

wr7:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ----- to sprue (as metal_flow_system) ----- *)

wr8:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));
  END_ENTITY;
(*

```

### Formal propositions:

**WR1:** There shall be exactly one reference to attribute **description** of either 'static', 'tilt' or 'low pressure'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'number of impressions'.

**WR3:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type **die\_mould\_vent**.

**WR4:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type an **ejector\_system**.

**WR5:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type a **production\_die\_mould**.

**WR6:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type an **ingate**.

**WR7:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type an **runner**.

**WR8:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** that is of type **applied\_action\_assignment** in which the **items** is a **shape\_aspect** of type a **sprue**.

### 5.2.3.1.180 pin\_center

A **pin\_center** is a type of **casting\_feature\_definition** that is the distance between the centers of two pins located on a centerline.

NOTE A **Pin\_center** is defined according to 4.2.229.

#### EXPRESS specification:

```

*)
ENTITY pin_center
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

  (* - ingate_contacts to connection_transition (as pattern_transition) - *)

  wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND
          (it.name = 'dimension')) ) = 1 )))=1;
END_ENTITY;
  (*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item**, **length\_measure\_with\_unit** and **qualified\_representation\_item** with a **name** of 'dimension'.

### 5.2.3.1.181 pin\_tip

A **pin\_tip** is a type of **casting\_feature\_definition** that is end of an ejector pin that intersects with the core cavity of the core box that forms the core.

NOTE A **Pin\_tip** is defined according to 4.2.230.

#### EXPRESS specification:

```
*)
ENTITY pin_tip
  SUBTYPE OF (ejector_design_feature);
  WHERE

(* ----- pin_tips to shape_element (as shape)----- *)

WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='pin tip occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pin tip shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) =1;
  END_ENTITY; -- pin_tip
(*
```

#### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'pin tip occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'pin tip shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.182 placed\_datum\_target\_feature

A **placed\_datum\_target\_feature** is a type of **datum\_target** that represents the implicit definition. See ARM definition for **Placed\_target** in paragraph 4.2.231 for more information.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.30 of ISO 10303-224:2006.

EXPRESS specification:

\*)

```

ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
  WHERE
wr1 : (SELF.description IN ['point','line','rectangle','circle']);
wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);

wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) )) |
  (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT'
  IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1)) ))
  = 0)) )) = 0);

wr4 : ((NOT (SELF.description = 'point')) OR
  (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( dtm_rep <*
  QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) )) |
  (NOT (SIZEOF(dtm_rep.used_representation.items) = 1)) ))
  = 0)) )) = 0));

wr5 : ((NOT (SELF.description IN ['line','circle'])) OR
  (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.' +
  'SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) )) |
  (NOT (SIZEOF(dtm_rep.used_representation.items) = 2)) ))
  = 0)) )) = 0));

```



```

wr6 : ((NOT (SELF.description = 'rectangle')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(dtm_rep.used_representation.items) = 3)) ))
= 0)) )) = 0));

wr7 : ((NOT (SELF.description = 'circle')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'target diameter')) ))
= 1)) )) = 0)) )) = 0));

wr8 : ((NOT (SELF.description = 'line')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
= 0)) )) = 0));

```

```

wr9 : ((NOT (SELF.description = 'rectangle')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) )
= 0)) ) = 0));

wr10: ((NOT (SELF.description = 'rectangle')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'target width'))))= 1))))
= 0)) )=0));
END_ENTITY; -- placed_datum_target_feature
(*

```

Formal propositions:

**WR1:** The **description** for the **placed\_datum\_target\_feature** shall be either 'point', 'line', 'rectangle' or 'circle'.

**WR2:** A **placed\_datum\_target\_feature** shall have exactly one implicit representation.

**WR3:** Exactly one **representation\_item** used for the representation of the **placed\_datum\_target\_feature** shall be of type **placement** with a **name** of 'orientation'.

**WR4:** If the **placed\_datum\_target\_feature** is a point, the **representation** shall contain exactly one **representation\_items** in its set of **items**.

**WR5:** If the **placed\_datum\_target\_feature** is a line or circle, the **representation** shall contain exactly two **representation\_items** in its set of **items**.

**WR6:** If the **placed\_datum\_target\_feature** is a rectangle, the **representation** shall contain exactly three **representation\_items** in its set of **items**.

**WR7:** If the **description** of the **placed\_datum\_target\_feature** is 'circle', exactly one **representation\_item** used for the implicit representation of the **placed\_datum\_target\_feature** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'target diameter'.

**WR8:** If the **description** of the **placed\_datum\_target\_feature** is 'line', exactly one **representation\_item** used for the implicit representation of the **placed\_datum\_target\_feature** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'target length'.

**WR9:** If the **description** of the **placed\_datum\_target\_feature** is 'rectangle', exactly one **representation\_item** used for the implicit representation of the **placed\_datum\_target\_feature** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'target length'.

**WR10:** If the **description** of the **placed\_datum\_target\_feature** is 'rectangle', exactly one **representation\_item** used for the implicit representation of the **placed\_datum\_target\_feature** shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'target width'.

### 5.2.3.1.183 process\_parameter\_record

A **process\_parameter\_record** is a type of **activity\_result\_record** that is a log of the value of a manufacturing variable during manufacture of a part.

NOTE A **Process\_parameter\_record** is defined according to 4.2.235.

#### EXPRESS specification:

```

*)
ENTITY process_parameter_record
  SUBTYPE OF (activity_result_record);
  WHERE

  (* - to property_parameter (as actual_parameter) ----- *)

wr1: (SIZEOF(QUERY( it <* SELF.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
  * TYPEOF(it)) = 1) AND
  (it.name = 'actual_parameter'))=1);

  (* -- to property_parameter (as planned_parameter) ----- *)

wr2: (SIZEOF(QUERY( it <* SELF.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
  * TYPEOF(it)) = 1) AND
  (it.name = 'planned_parameter'))=1);
  END_ENTITY;
  (*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'actual parameter'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'planned parameter'.

### 5.2.3.1.184 process\_plan\_activity

The **process\_plan\_activity** is a type of **action\_method\_with\_associated\_documents** which defines **action\_methods** required to define a process plan type of **action**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.77 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY process_plan_activity
  SUBTYPE OF (action_method);
  WHERE

(* ----- activity_number ----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'activity number')) ) =1 ))))=1;

(* ----- frequency ----- *)

wr2:  (SIZEOF(QUERY(ar <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE' ) |
  (ar.name='frequency') )) <=1);

(* ---- to activity_execution_result to activity (as execution_result) - *)

wr3:  (SIZEOF(QUERY(ar <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD' ) |
  ('CAST_PARTS_SCHEMA.ACTIVITY_EXECUTION_RESULT'
  IN TYPEOF(ar) )) ) <=1);

```

```

(* ----to organization (as performed_by_organization_id) ----- *)

wr4: (SIZEOF(QUERY ( aam <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
  ((aam\organization_assignment.role.name='activity organization id') AND
  ('CAST_PARTS_SCHEMA.ORGANIZATION'
  IN TYPEOF(aam\organization_assignment.assigned_organization))))<=1);

(* ----- to special_instruction (as activity_information) ----- *)

wr5:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'special instruction')
  )))) >=0 ))))>=1;

(* --- to supplemental_document (as process_specification) -----*)

wr6: (NOT (('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF))) OR
  (SIZEOF(QUERY ( adr <*
  SELF\action_method_with_associated_documents.documents |
  (('CAST_PARTS_SCHEMA.PROCESS_PLAN_SPECIFICATION'
  IN TYPEOF(adr)) AND
  (adr.kind.product_data_type = 'supplemental documents')))) >= 0);

(* ----- activity to performance_rate (as duration) ----- *)

wr7:  (NOT(SIZEOF(QUERY(ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.ALLOWED_TIME',
    'CAST_PARTS_SCHEMA.PRODUCTION_RATE'] * TYPEOF(ap)) >= 0)))=0));

(* --activity to design_reference (as graphics_representation) -----*)
(* --activity to view_reference (as graphics_representation) -----*)
(* --activity to illustration (as graphics_representation) -----*)

wr8: (NOT('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) OR
  (NOT (SIZEOF(QUERY( doc <* SELF.documents |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.DESIGN_REFERENCE',
    'CAST_PARTS_SCHEMA.VIEW_REFERENCE',
    'CAST_PARTS_SCHEMA.ILLUSTRATION'] * TYPEOF(doc)) >=0 ))) =0 )));

```

ISO 10303-223:2008 (E)

```
(* ----- activity to cast_part (as output_item) ----- *)

wr9: (SIZEOF(QUERY ( apr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD' ) |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(apr)) AND
(NOT(SIZEOF(QUERY(it <* apr.items |
(SIZEOF([
'CAST_PARTS_SCHEMA.CAST_PART',
'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING'] * TYPEOF(it)) <=1 )
)) =0 )) )) )<=1);

(* -- activity to generic_manufacturing_resource (as consumable_resource) *)

wr10: (SIZEOF(QUERY(ar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE' ) |
('CAST_PARTS_SCHEMA.GENERIC_MANUFACTURING_RESOURCE' IN
TYPEOF(ar)))>=0);
END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_ - representation\_item** and **count\_measure** with a **name** of 'activity number'.

**WR2:** Exactly one reference that is the **usage** in exactly one **action\_resource** with **name** of 'frequency'.

**WR3:** There shall be exactly one reference from an **action** through the **chosen\_method** attribute that is of type **activity\_execution\_result**.

**WR4:** There shall exactly one references from a **applied\_organization\_assignment** through the **items** of type **organization\_assignment** that has one reference of **role** that is an **organization\_role** with **name** of 'activity organization id' that references an **organization** through the **assigned\_organization**.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'special instruction'.

**WR6:** There shall be zero or one **document** with a **name** of 'supplemental documents' referenced through **documents**.

**WR7:** There shall be zero or more references by a **production\_rate** or **allowed\_time** of kind **action\_ - property** through **definition**.

**WR8:** There shall be zero or one **design\_reference**, **view\_reference** or **illustration** referenced through **documents**.

**WR9:** There shall be a reference through the **assigned\_action\_method** attribute by **action\_method\_assignment** of type **applied\_action\_method\_assignment** that references through the **items** attribute **product\_definition** of type **cast\_part** or **cast\_part\_with\_rigging**.

**WR10:** Exactly one reference that is the **usage** in exactly one **action\_resource** is of type **generic\_manufacturing\_resource**.

### 5.2.3.1.185 process\_plan\_security

The **process\_plan\_security** is a type of **security\_classification** that assigns a security level to an to a **process\_plan\_version** and a **process\_plan\_activity**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.78 of ISO 10303-240:2005.

#### EXPRESS specification:

```
*)
  ENTITY process_plan_security
  SUBTYPE OF (SECURITY_CLASSIFICATION);
  WHERE

  (* ----- classification_date ----- *)

wr1: SIZEOF(QUERY ( da <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.DATE_ASSIGNMENT'
  IN TYPEOF(da)) AND
  ('CAST_PARTS_SCHEMA.CALENDAR_DATE'
  IN TYPEOF(da.assigned_date))) AND
  (da.role.name = 'classification date'))))
  = 1;

  (* ----- declassification_date ----- *)

wr2: SIZEOF(QUERY ( da <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da))
  AND ('CAST_PARTS_SCHEMA.CALENDAR_DATE'
  IN TYPEOF(da.assigned_date))) AND
  (da.role.name = 'declassification date'))))
  = 1;
```

ISO 10303-223:2008 (E)

```
(* ----- to activity ( as identified_by_activity) ----- *)

wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.SECURITY_CLASSIFICATION_ASSIGNMENT.' +
  'ASSIGNED_SECURITY_CLASSIFICATION') |
  (('CAST_PARTS_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
  IN TYPEOF(ap)) AND
  (NOT (SIZEOF(QUERY ( it <* ap.items |
  ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(it))))
  = 1)))) = 0;

(* -----to process_plan_version (as identified_by_process_plan) ----- *)

wr4:SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.' +
  'SECURITY_CLASSIFICATION_ASSIGNMENT.ASSIGNED_SECURITY_CLASSIFICATION') |
  (('CAST_PARTS_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
  IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items |
  ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(it))))
  = 1)))) = 0;

END_ENTITY;
(*
```

#### Formal propositions:

**WR1:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'classification data' through **role**, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR2:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'declassification data' through **role**, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR3:** There shall be exactly one reference by a **security\_classification\_assignment** through **assigned\_security\_classification** that is a kind of **applied\_security\_classification\_assignment** that references a **process\_plan\_activity** through **items**.

**WR4:** There shall be exactly one reference by a **security\_classification\_assignment** through **assigned\_security\_classification** that is a kind of **applied\_security\_classification\_assignment** that references a **process\_plan\_version** through **items**.

### 5.2.3.1.186 process\_plan\_specification

The **process\_plan\_specification** is a type of **document** that defines the subclass and revision data for process plan specifications.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.79 of ISO 10303-240:2005.



EXPRESS specification:

\*)

```

ENTITY process_plan_specification
  SUBTYPE OF (document);
  WHERE
  wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    (duc.subject_element = 'revision') )) = 1;
  wr2: SIZEOF(QUERY ( duc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
    (duc.subject_element = 'subclass') )) = 1;
END_ENTITY; -- process_plan_specification
(*)

```

Formal propositions:

**WR1:** There shall be a reference from **document\_usage\_constraint** through the **source** attribute that has an attribute with a **subject\_element** of 'revision'.

**WR2:** There shall be a reference from **document\_usage\_constraint** through the **source** attribute that has an attribute with a **subject\_element** of 'subclass'.

**5.2.3.1.187 process\_plan\_version**

A **process\_plan\_version** is a type of **product\_definition\_process** that is the collection of instructions, revisions, manufacturing activities and processes, along with their sequence of execution, that is required to machine a specific part.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.80 of ISO 10303-240:2005.

EXPRESS specification:

\*)

```

ENTITY process_plan_version
  SUBTYPE OF (product_definition_process);
  WHERE
  wr1: (NOT(SIZEOF(QUERY ( ppp <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.PROCESS') |
    (SIZEOF(['CAST_PARTS_SCHEMA.DESIGN_PART',
      'CAST_PARTS_SCHEMA.CAST_PART',
      'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
      'CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION']
      * TYPEOF(ppp.defined_product)
      )=1) ))=0));

```

ISO 10303-223:2008 (E)

```

wr2: (SIZEOF(QUERY(act <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATING_ACTION') |
('CAST_PARTS_SCHEMA.ACTIVITY_EXECUTION_RESULT'
IN TYPEOF(act.related_action)
))=1);

(* process_plan to casting_activity (as activities) *)

wr3: (SIZEOF(QUERY ( duc <* USEDIN(SELF.chosen_method,
'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
('CAST_PARTS_SCHEMA.CASTING_ACTIVITY' IN TYPEOF(duc.related_method)
)))=1);

(* to manufacturing_process (as activities to produce part) *)

wr4: (NOT(SELF\action.id = 'mini plan')) AND
(NOT(SIZEOF(QUERY ( duc <* USEDIN(SELF.chosen_method,
'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
(('CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP' IN TYPEOF(duc)) AND
( SIZEOF([
'CAST_PARTS_SCHEMA.NON_MACHINING_PROCESS',
'CAST_PARTS_SCHEMA.MACHINING_PROCESS']
* TYPEOF(duc.related_method)=1))))=0));

(* ----- to material_usage as amount of raw materials) ----- *)

wr5:(NOT(SELF\action.id = 'mini plan')) AND
(NOT(SIZEOF(QUERY( act <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_ASSIGNMENT' IN TYPEOF(act)) AND
(SIZEOF(QUERY( mu <* act.items |
'CAST_PARTS_SCHEMA.MATERIAL_USAGE' IN TYPEOF(mu))=1))))=0));

(* ----- to ordered_part (quantity_of_parts) ----- *)

wr6:(NOT(SELF\action.id = 'mini plan')) AND
(NOT (SIZEOF(QUERY( act <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ORDERED_PART.ITEMS') |
(SIZEOF(QUERY ( pd <* USEDIN(act,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'quantity'))
)) =1 )) )) =0 )) )) =0 ))) =0));

```

```
(* to property_parameter (as process plan properties) *)

wr7: (NOT(SELF\action.id = 'mini plan')) AND
      (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
        (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
        (SIZEOF(QUERY ( it <* apr.representation.items |
        ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
        * TYPEOF (it) )=1 ) AND
        (it.name='process plan properties' ))
        ))=1)))=0))))=0));

(* to special_instruction (as process plan information) *)

wr8: (NOT(SELF\action.id = 'mini plan')) AND
      (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
        (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
        (SIZEOF(QUERY ( it <* apr.representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'special instruction'))
        )) = 1))) = 0)))) = 0));
END_ENTITY;
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **process** in exactly one **process\_product\_association** in which the **defined\_product** is a **product\_definition** in which the **defined\_product** attribute references a **design\_part**, **cast\_part**, **\_cast\_part\_with\_rigging**, or a **casting\_product\_definition**.

**WR3:** There shall be exactly one reference through the **chosen\_method** attribute to a **action\_method** that is reference by an **action\_method\_relationship** through the **relating\_method** that references through the **related\_method** attribute an **action\_method** of type **casting\_activity**.

**WR4:** If the attribute **id** is not 'mini plan' then there shall be exactly one reference through the **chosen\_method** attribute to a **action\_method** that is reference by an **action\_method\_relationship** through the **relating\_method** that references through the **related\_method** attribute an **action\_method** of type **machining\_process** or **non\_machining\_process**.

**WR5:** If the attribute **id** is not 'mini plan' then there shall be exactly one reference through the **assigned\_attribute** of an **action\_assignment** that is of type **applied\_action\_assignment** that references through the **items** attribute a **property\_definiton** of type **material\_usage**.

**WR6:** If the attribute **id** is not 'mini plan' then there shall be exactly one reference by an **ordered\_part** through the **items** attribute that is reference by a **property\_definition** that is referenced through the **definition** attribute that is referenced by a **property\_definition\_representation** through the **definition** attribute that referenced through the **used\_representation** a **representation** that references through the **items** attribute a **measure\_representation\_item** with a **name** of 'quantity'.

**WR7:** If the attribute **id** is not 'mini plan' then there shall be exactly one reference that is a **representation** in which exactly one **items** is a **measure\_representation\_item** or **descriptive\_representation\_item** with **name** of 'process plan properties'.

**WR8:** Exactly one reference that is a **representation** in which exactly one **items** is a **descriptive\_representation\_item** with a **name** of 'special instruction'.

### 5.2.3.1.188 process\_requirement

A **process\_requirement** is a type of **customer\_casting\_requirement** that is an activity or operation that is required to produce the cast part.

NOTE A **Process\_requirement** is defined according to 4.2.240.

#### EXPRESS specification:

\*)

```
ENTITY process_requirement
    SUBTYPE OF (customer_casting_requirement);
WHERE
```

```
(* ----- to descriptive_parameter(as process_name) ----- *)
```

```
wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'process name')
    ) ) =1 ))=0)));
```

```
END_ENTITY;
```

```
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'process name'.

### 5.2.3.1.189 production\_core\_box

A **production\_core\_box** is a type of **production\_tool** that defines the structure that has the shape of the desired core to be produced.

NOTE A Production\_core\_box is defined according to 4.2.242.

#### EXPRESS specification:

```

*)
ENTITY production_core_box
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ***** box_type ***** *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'core box type')))) =1 )))=1;

(* ***** vent_type ***** *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'core vent type')))) =1 )))=1;

(* ----- core_box_material ----- *)

WR3: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN ['wood','cast iron','urethane','aluminum'])) ) ) = 1;

(* ----- number_of_impressions ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of impressions')) ) ) =1 )))=1;

```

ISO 10303-223:2008 (E)

```
(* -----shrink_factor----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor')) ) =1 ))))=1;

(* -----production_core_box to casting_insert (as inserts) --- *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'casting insert reference usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_INSERT'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* ---production_core_box to chaplet_pad (as chaplet_pad_definition) --- *)

wr7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chaplet pad reference usage') AND
  ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* -----production_core_box to core (as core_definiton) ----- *)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core reference usage') AND
  ('CAST_PARTS_SCHEMA.CORE' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 ))) >=1;
```

```
(* ---- production_core_box to core_equipment (as core_made_by )----- *)
```

```
wr9: (NOT (SIZEOF (QUERY ( adr <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.ACTION_METHOD'
  IN TYPEOF(adr.assigned_action_method)) AND
  (SIZEOF (QUERY (ar <* USEDIN (adr.assigned_action_method,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('CAST_PARTS_SCHEMA.CORE_EQUIPMENT' IN TYPEOF(ar) ))) >=1)))) =0));
```

```
(* -production_core_box to core_print (as core_print_definition) ----- *)
```

```
wr10: SIZEOF (QUERY ( sa <* get_property_definition_shape_aspect (SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF (QUERY (sar <* USEDIN (sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print reference usage') AND
  ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 ))) >=1;
```

```
(* ----- production_core_box to drafted_surface (as draft) *)
```

```
wr11: SIZEOF (QUERY (pdr <* get_property_definition_representations (SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ))) >=1;
```

```
(* -production_core_box to loose_piece (as loose_piece_requirement) --- *)
```

```
wr12: SIZEOF (QUERY ( sa <* get_property_definition_shape_aspect (SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF (QUERY (sar <* USEDIN (sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece reference usage') AND
  ('CAST_PARTS_SCHEMA.LOOSE_PIECE' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 ))) >=1;
```

```
(* production_core_box to in_facility_location (as core_box_location) *)
```

```
wr13: ((SIZEOF (QUERY (pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ))) =1) OR
  (SIZEOF (QUERY (ada <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='core box stored')) =1)
  ));
```

ISO 10303-223:2008 (E)

```
(* ----- production_core_box to item_size (as guide_pin) ----- *)

WR14: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'guide pin size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 ))) >=1;

(* - production_core_box to item_size (as outside_shape_of_core_box) --- *)

WR15: ((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'outside size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1)
OR
(SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'outside size usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1));

(* - production_core_box to parting_surface (as cope_and_drag_parting) *)

wr16:SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='cope and drag parting')))) >=1;

(* - production_core_box to parting_surface (as offset_parting) --- *)

wr17: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='offset parting')))) >=1;
```



```
(* - production_core_box to planar_element(as vent_placement) ----- *)

WR18: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment area shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;

(* --- production_core_box to shape_element (as impression_shape)----- *)

WR19: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'impression shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;

(* * to machining_allowance (as machine_stock_on_tool) ** *)

wr20: (SIZEOF(QUERY (co <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))>=1);

END_ENTITY; -- production_core_box
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core box type'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'core vent type'.

**WR3:** Exactly one reference that is the **definition** in exactly one **material\_designation** with **name** of 'wood', 'cast iron', 'urethane' or 'aluminum'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of impressions'.

**WR5:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'shrink factor'.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'casting insert reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **casting\_insert**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'productio core box reference occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chaplet pad reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chaplet\_pad**.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'productio core box reference occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **core**.

**WR9:** Exactly one reference that is the **items** in exactly one **applied\_action\_method\_assignment** of type **action\_method\_assignment** which the **assigned\_action\_method** was an **action\_method** that has one reference that is the **usage** in exactly one **action\_resource** of type **core\_equipment**.

**WR10:** Exactly one **shape\_aspect** with a **description** of 'productio core box reference occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core print reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **core\_print**.

**WR11:** Exactly one **representation** used for the implicit representation shall be of type **drafted\_surface** with a **name** of 'draft'.

**WR12:** Exactly one **shape\_aspect** with a **description** of 'productio core box reference occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'loose piece reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **loose\_piece**.

**WR13:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the **items** attribute that is of type **person\_and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_role** with **name** of 'core box stored'.

**WR14:** Exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'guide pin size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR15:** Exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'production core box size usage' in which the **relating\_shape\_aspect** is a **shape\_aspect** or exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'production core box size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR17:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'cope and drag parting'.

**WR18:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'offset parting'.

**WR19:** Exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'investment area shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR20:** Exactly one **shape\_aspect** with a **description** of 'production core box occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'impression shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR21:** Exactly one reference by a **property\_definition** through the **definition** attribute and the **property\_definition** is of type **machine\_allowance**.

### 5.2.3.1.190 production\_die\_cast\_mould

A **production\_die\_cast\_mould** is a type of **production\_tool** that defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces.

NOTE A **Production\_die\_cast\_mould** is defined according to 4.2.243.

EXPRESS specification:

\*)

```

ENTITY production_die_cast_mould
  SUBTYPE OF (production_tool);
  WHERE

  (* ***** port_connection_type ***** *)

  wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'port connection type')))) =1 )))=1;

  (* ***** insert_material_type ***** *)

  wr2: (SIZEOF(QUERY (co <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS' ) |
    'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))<=1);

  (* ----- cooling_requirement ----- *)

  wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION' ) |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) ) |
    (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'cooling requirement' )
    AND (it.description IN ['true','false'] )
    )))=1))))=0))))=0);
  
```

```

(* ----- die_type ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'die type') AND
    ((it.description IN ['hot chamber','cold chamber','vacum chamber']
    )) )) ) =1 )) )>=1;

(* ----- inserts ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'insert') AND
    ((it.description IN ['unit','combination','cast in','none']
    )) )) ) <=1 )) )>=1;

(* production_die_cast_mould to item_size (as external_box_dimensions) *)

WR6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die cast mould dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

(* ***** production_die_cast_mould to sprue (as pouring_sprue) ***** *)

wr7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue reference usage') AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

```

```

(* ***** production_die_cast_mould to runner (as gate_runner) ***** *)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gate runner reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.RUNNER',
  'CAST_PARTS_SCHEMA.INGATE']
  * TYPEOF(sar.relatng_shape_aspect)) =1))) >=1 ))))>=1;

(* production_die_cast_mould to slide (as movable_die_items) ***** *)
(* ** production_die_cast_mould to die_clamping (as movable_die_items)
*)

wr9: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'movable die clamping reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.SLIDE',
  'CAST_PARTS_SCHEMA.DIE_CLAMPING']
  * TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 ))))>=1;

(* *** production_die_cast_mould to cooling_port (as cooling_type) *****
*)

wr10: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'cooling port reference usage') AND
  ('CAST_PARTS_SCHEMA.COOLING_PORT' IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 )))) >=1;

(* to secondary_tooling (as secondary_tooling_requirements) ***** *)

wr11: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'secondary tooling reference usage') AND
  ('CAST_PARTS_SCHEMA.SECONDARY_TOOLING'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

```

```

(* production_die_cast_mould to die_clamping (as stationary_die_items) *)
(* production_die_cast_mould to shot_sleeve (as stationary_die_items) *)

wr12: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production die cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'stationary die clamping reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.SHOT_SLEEVE',
  'CAST_PARTS_SCHEMA.DIE_CLAMPING']
  * TYPEOF(sar.relatinq_shape_aspect)) =1))) >=1 )))>=1;

END_ENTITY; -- production_die_cast_mould
(*

```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'port connection type'.

**WR2:** Exactly one reference that is the **definition** in exactly one **material\_designation**.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'cooling requirement' and a **description** of either 'true' or 'false'.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'die type' and a **description** of either 'hot chamber' or 'cold chamber'.

**WR5:** Exactly one reference that is the **related\_product\_definition** in exactly one **product\_definition\_relationship** in which the **relating\_product\_definition** is an **production\_die\_mould** which has exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'inserts' and a **description** of 'unit', 'combination', 'cast in' or 'none'.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'production die cast mould dimension usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'sprue reference usage' in which the **relating\_shape\_aspect** is an **sprue**.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'runner reference usage' in which the **relating\_shape\_aspect** is an **runner** or **ingate**.

**WR9:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'slide reference usage' in which the **relating\_shape\_aspect** is an **slide** or **die\_clamping**.

**WR10:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'cooling port reference usage' in which the **relating\_shape\_aspect** is an **cooling\_port**.

**WR11:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'secondary tooling reference usage' in which the **relating\_shape\_aspect** is an **secondary\_tooling**.

**WR12:** Exactly one **shape\_aspect** with a **description** of 'production die cast mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'die clamping reference usage' in which the **relating\_shape\_aspect** is an **die\_clamping** or **shot\_sleeve**.

### 5.2.3.1.191 production\_die\_mould

A **production\_die\_mould** is a type of **production\_tool** that defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces.

NOTE A **Production\_die\_mould** is defined according to 4.2.244.



EXPRESS specification:

```

*)
ENTITY production_die_mould
  SUBTYPE OF (production_tool);
  WHERE

(* ***** die_type ***** *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'die type')))) =1 ))))=1;

(* --- production_die_mould to item_size (as external_box_dimensions) -*)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relater_shape_aspect))
  )) =1 )))) >=1;

(* ***** production_die_mould to slide (as slide_requirement) ***** *)

wr3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'slide reference usage') AND
  ('CAST_PARTS_SCHEMA.SLIDE' IN TYPEOF(sar.relater_shape_aspect))
  )) >=0 )))) >=1;

(* --- production_die_mould to gating_system (as gate_rigging) -----
*)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gating system reference usage') AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM'
  IN TYPEOF(sar.relater_shape_aspect))
  )) =1 )))) >=1;

```

ISO 10303-223:2008 (E)

```
(* ----- production_die_mould to die_core (as core for die) -----
*)
(* ---- production_die_mould to core(as core for die) ----- *)

wr5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.CORE',
  'CAST_PARTS_SCHEMA.DIE_CORE']
  * TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 )))>=1;

END_ENTITY; -- production_die_mould
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'die type'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'production die mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'production die mould dimension usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'production die mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'slide reference usage' in which the **relating\_shape\_aspect** is an **slide**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'production die mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'gating system reference usage' in which the **relating\_shape\_aspect** is an **gating\_system**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'production die mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **core** or **die\_core**.

### 5.2.3.1.192 production\_investment\_cast\_mould

A **production\_investment\_cast\_mould** is a type of **production\_tool** that defines the revised geometric shape, dimensional information, cores required, draft required, and parting surfaces.

NOTE A **Production\_investment\_cast\_mould** is defined according to 4.2.245.

#### EXPRESS specification:

```

*)
ENTITY production_investment_cast_mould
  SUBTYPE OF (production_tool);
  WHERE

(* * production_investment_cast_mould to slide (as slide_requirement) * *)

wr1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production investment cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'slide reference usage') AND
  ('CAST_PARTS_SCHEMA.SLIDE' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=1 ))) >=1;

(* ----- to in_facility_location (as die_mould_location) *)

wr2: ((SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) )) ) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='core box stored'))=1)
  );

END_ENTITY; -- production_investment_cast_mould
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'shrink factor'.

**WR2:** Either exactly one reference by a **property\_definition** through the **definition** attribute that is of type **product\_definition\_shape** and is referenced by a **property\_definition\_relationship** through the **definition** attribute that is of type **shape\_definition\_representation** and references through the **used\_representation** a **representation** that is of type **in\_facility\_location** or exactly one reference by an **applied\_person\_and\_organization\_assignment** through the **items** attribute that is of type **person\_-and\_organization\_assignment** that references through the **role** attribute a **person\_and\_organization\_-role** with **name** of 'tooling location'.

### 5.2.3.1.193 production\_tool

A **production\_tool** is a type of **casting\_product\_definition** that is a mould which contain die cavities of the design shape required to produce production casting of a designed part for a customer.

NOTE A **Production\_tool** is defined according to 4.2.246.

EXPRESS specification:

\*)

```
ENTITY production_tool
  SUBTYPE OF (die_design_feature);
  WHERE
```

(\* -----shrink\_factor----- \*)

```
wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor')) ) ) =1 )))=1;
```

(\* ----- male\_or\_female\_mould----- \*)

```
wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'male or female') AND
  ((it.description IN ['male mould','female mould'] ) ) ) ) =1 ) ) )>=1;
```

(\* ----- number\_of\_cavities ----- \*)

```
wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of cavities')) ) ) =1 )))=1;
```

```

(* ----- production_tool to item_size (as guide_pin) ----- *)

WR4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'guide pin shape usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) <=1 ))) >=1;

(* ----- production_tool to die_core (as core_for_die) ----- *)

wr5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'die core reference usage') AND
  ('CAST_PARTS_SCHEMA.DIE_CORE' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* -- production_tool to ejector_system (as ejection_definition) ---- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector system reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) =1 ))) >=1;

(* - production_tool to production_core_box (as core_definition) --- *)

wr7:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production core box reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* * to machining_allowance (as allowance_on_machinable_surfaces) ** *)

wr8: (SIZEOF(QUERY (co <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))<=1);

```

ISO 10303-223:2008 (E)

```
(* - production_tool to parting_surface (as male_or_female_parting) - *)

wr9: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
    IN TYPEOF ( pdr.used_representation ) ) AND
    (pdr.used_representation.name='male or female parting')))) >=1;

(* - production_tool to parting_surface (as offset_parting_surfaces) *)

wr10: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
    IN TYPEOF ( pdr.used_representation ) ) AND
    (pdr.used_representation.name='offset parting')))) >=0;

(* production_tool to drafted_surface (as draft) *)

wr11: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
    IN TYPEOF ( pdr.used_representation ) ))) >=1;

(* ----- to shape_aspect (as outside_shape_of_die_mould)----- *)

(* ----- production_tool to (as outside_shape_of_die_mould) -----
-- *)

WR12: ((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production tool outside shape usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relating_shape_aspect))
  )) =1 )))) >=1)
OR
(SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production tool outside shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relating_shape_aspect))
  )) =1 )))) >=1));
```

```

(* --- production_tool to shape_element (as mould_impression_shape)-----
*)

WR13:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production tool impression shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relating_shape_aspect))
  )) >=1 )))) >=1;

(* ----- production_tool to riser (as riser_on_mould) ----- *)
(* ---- production_tool to riser_contact (as riser_on_mould) ----- *)

wr14: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.RISER',
  'CAST_PARTS_SCHEMA.RISER_CONTACT']
  * TYPEOF(sar.relating_shape_aspect)) =1))) >=0 ))))>=1;

(* ----- production_tool to vent ( as vent requirement) -----
- *)

WR15: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'vent reference usage') AND
  ('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

END_ENTITY; -- production_tool
(*

```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'male or female' and **description** of 'shrink factor'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'male or female' and **description** of 'male or female' and **description** of 'male mould','female mould'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of cavities'.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'guide pin reference usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'die core reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **die\_core**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'ejector system reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **ejector\_system**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production core box reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_core\_box**.

**WR8:** Exactly one reference of **definition** is a **property\_definition** of type **machining\_allowance**.

**WR9:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'male and female parting'.

**WR10:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'offset parting surface'.

**WR11:** Exactly one **representation** used for the implicit representation shall be of type **drafted\_surface**.

**WR12:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'production tool outside shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect** or exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'production tool outside shape usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.



**WR13:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'production tool impression shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR14:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'riser contact reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **riser** or **riser\_contact**.

**WR15:** Exactly one **shape\_aspect** with a **description** of 'production tool occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'vent reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **vent**.

### 5.2.3.1.194 production\_pattern\_definition

A **production\_pattern\_definition** is a type of **casting\_product\_definition** that defines the geometric shape, dimensional information, cores required, draft required, and parting surfaces.

NOTE A **Production\_pattern\_definition** is defined according to 4.2.247.

#### EXPRESS specification:

\*)

```
ENTITY production_pattern_definition
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
```

```
(* ***** condition ***** *)
```

```
WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'condition') AND
  (it.description IN ['good', 'bad', 'fair', 'new'] )) )) ) =1 ))) >=1;
```

ISO 10303-223:2008 (E)

```
(* ----- number_of_patterns_mounted ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
      (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
      ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'number of patterns mounted')) ) =1 )))=1;

(* -----shrink_factor----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it) = 2) AND
        (it.name = 'shrink factor')) ) =1 )))=1;

(* ---production_pattern_definition to casting_insert (as inserts) -- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'casting insert reference usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_INSERT'
      IN TYPEOF(sar.relatinq_shape_aspect))
    )) >=0 ))) >=1;

(* -- to chaplet_pad (as chaplet_pad_definition) -- *)

wr5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'chaplet pad reference usage') AND
    ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relatinq_shape_aspect))
    )) >=0 ))) >=1;
```

```

(* --production_pattern_definition to chill (as chill_pattern) --- *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chill reference usage') AND
  ('CAST_PARTS_SCHEMA.CHILL' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* -- to in_facility_location (as pattern_location) *)

wr7: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELf) |
  ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) )) ) =1;

(* -- to production_core_box(as cores_for_pattern) --*)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production core box reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* -- to core_print (as core_print_definition) -- *)

wr9: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print reference usage') AND
  ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* ---- production_pattern_definition to drafted_surface (as draft) *)

wr10: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELf) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) )) ) >=1;

```

ISO 10303-223:2008 (E)

```
(* - to loose_piece(as loose_piece_requirement) -- *)

wr11: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece reference usage') AND
  ('CAST_PARTS_SCHEMA.LOOSE_PIECE' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* -- to parting_surface (as cope_and_drag_parting) - *)

wr12: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='cope and drag parting')))) =1;

(* -- to parting_surface (as offset_parting) ----- *)

wr13: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='offset parting')))) >=0;

(* -- to pattern_plate (as match_plate_patterns) ---- *)

wr14: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pattern plate reference usage') AND
  ('CAST_PARTS_SCHEMA.PATTERN_PLATE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* ----- to riser (as riser_on_pattern) ----- *)
(* ----- to riser_contact (as riser_on_pattern) ----- *)

wr15: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser reference usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.RISER',
  'CAST_PARTS_SCHEMA.RISER_CONTACT']
  * TYPEOF(sar.relatinq_shape_aspect)) =1))) >=0 ))))>=1;
```

```
(* ----- to shape_element (as invest_area)----- *)
WR16:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production pattern definition shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;
END_ENTITY; -- production_pattern_definition
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'condition' and a **description** of either 'good', 'bad', 'fair' or 'new'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of patterns mounted'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'shrink factor'.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'casting insert reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **casting\_insert**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chaplet pad reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chaplet\_pad**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chill reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chill**.

**WR7:** Exactly one **representation** used for the implicit representation shall be of type **in\_facility\_location**.

**WR8:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production core box reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **production\_core\_box**.

**WR9:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'core print reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **core\_print**.

**WR10:** Exactly one **representation** used for the implicit representation shall be of type **drafted\_surface**.

**WR11:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'loose piece reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **loose\_piece**.

**WR12:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'cope and drag parting'.

**WR13:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface** with a **name** of 'offset parting surface'.

**WR13:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** of type **casting\_feature\_component\_relationship** in which the **relating\_product\_definition** is a **pattern\_plate**.

**WR14:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'pattern plate reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **pattern\_plate**.

**WR15:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'riser reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **riser** or **riser\_contact**.

**WR16:** Exactly one **shape\_aspect** with a **description** of 'production pattern definition occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production pattern definition shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.195 production\_rate

A **production\_rate** is an **action\_property** which is the time it takes to produce a measurable amount of work.

NOTE 1 A **Production\_rate** is defined according to 4.2.248.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.81 of ISO 10303-240:2005.

#### EXPRESS specification:

```

*)
  ENTITY production_rate
    SUBTYPE OF (action_property);
  WHERE

  wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF
        (it)) = 2) AND (it.name = 'time per unit')))) = 1))) = 1;

  wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'unit quantity'))))=1)))=1;

  wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'production rate source'))))=1)))=1;
  END_ENTITY;
  (*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the representation of an **production\_rate** shall be of type **measure\_representation\_item** and **time\_measure\_with\_unit** with a **name** of 'time per unit'.

**WR2:** Exactly one **representation\_item** used for the representation of an **production\_rate** shall be of type **measure\_representation\_item** with a name of 'unit quantity'

**WR3:** Exactly one **representation\_item** used for the representation of an **production\_rate** shall be of type **descriptive\_representation\_item** with a **name** of 'production rate source'.

### 5.2.3.1.196 production\_welding

A **production\_welding** is a type of **manufacturing\_activity** that is a type of **Non\_casting\_operation** and an added value operation that joins two or more cast parts or a cast part to wrought materials by a welding procedure before any machining is performed.

NOTE A **Production\_welding** is defined according to 4.2.249.

EXPRESS specification:

\*)

```
ENTITY production_welding
  SUBTYPE OF (non_casting_activity);
END_ENTITY;
```

(\*

### 5.2.3.1.197 property\_inspection

A **property\_inspection** is a type of **inspection\_or\_test\_result** that is the outcome of a measurement and comparing the resultant properties to the design requirememnt.

NOTE A **Property\_inspection** is defined according to 4.2.252.

EXPRESS specification:

\*)

```
ENTITY property_inspection
  SUBTYPE OF (inspection_or_test_result);
WHERE
```



```
(* --to property_parameter (as property_result)--- *)

wr1: (NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY( it <* pdr.used_representation.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
  * TYPEOF(it)) = 1) AND
  (it.name = 'property result')) =1 ))) =0));

(* --to property_parameter (as requirements)---- *)

wr2: (NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY( it <* pdr.used_representation.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
  * TYPEOF(it)) = 1) AND
  (it.name = 'requirements') ) ) >=1))) =0));

END_ENTITY; -- property_inspection
(*
```

#### Formal propositions:

**WR1:** Exactly one reference that is a **representation** in which exactly one **items** is a **measure\_ - representation\_item** or **descriptive\_representation\_item**.

**WR2:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_ - definition\_relationship** in which the **relating\_property\_definition** is an **property\_requirement**.

### 5.2.3.1.198 property\_relationship

A **property\_relationship** is a type of **representation** that is the allowable pairs of values for two variables.

NOTE A **Property\_relationship** is defined according to 4.2.257.

#### EXPRESS specification:

```
*)
ENTITY property_relationship
  SUBTYPE OF (representation);
  WHERE

(* --- x property name----- *)

wr1: SIZEOF(QUERY ( mri <* SELF.items | ((
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  (mri.name = 'x property name')) ) ) = 1;
```

ISO 10303-223:2008 (E)

```
(* --- y property name----- *)

wr2: SIZEOF(QUERY ( mri <* SELF.items | ((
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  (mri.name = 'y property name'))) ) = 1;

(* --- property_relationship to data_curve (as data_points)----- *)

wr3: SIZEOF(QUERY ( rep <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (('CAST_PARTS_SCHEMA.DATA_CURVE' IN TYPEOF(rep.rep_2)) AND
  (rep.rep_2.name='relationship curve')))) >= 1;

  END_ENTITY; -- property_relationship
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** with a **name** of 'x property name'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** with a **name** of 'y property name'.

**WR3:** Exactly one reference of **items** is a **data\_curve** with **name** of 'relationship curve'.

### **5.2.3.1.199 red\_line\_part**

A **red\_line\_part** is a type of **product\_definition** that identifies design inconsistencies for the part version being manufactured and may require engineering design changes to produce the cast part.

NOTE A **Red\_line\_part** is defined according to 4.2.261.

EXPRESS specification:

```

*)

ENTITY red_lined_part
  SUBTYPE OF (product_definition);
  WHERE

(* ----- red_lined_part to feature (as mark_up_feature) *)

wr1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='red lined part occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'red lined feature usage') AND
  ( SIZEOF(['CAST_PARTS_SCHEMA.INSTANCED_FEATURE',
  'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE']
  * TYPEOF(sar.relating_shape_aspect)) =1))) >=0 )))>=1;

(* ----- red_lined_part to shape_aspect (as mark_up_shape) *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='red lined part occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'red lined shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 ))) >=1;

(* ----- red_lined_part to material (as mark_up_material) *)

WR3: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='mark up material') AND
  (SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
  ))>=0))))=0));

(* ----- red_lined_part to property (as mark_up_property) *)

wr4: SIZEOF(QUERY(pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (pd.id='mark up property')) >=0;

  END_ENTITY; -- red_lined_part
(*)

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'red lined part occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'red lined feature usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **instanced\_feature** or a **casting\_instanced\_feature**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'red lined part occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'red lined shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR3:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'material definition' and of type **make\_from\_usage\_option** in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR4:** There shall be exactly one reference by a **property\_definition** through the **definition** attribute with an **id** of 'mark up property'.

### 5.2.3.1.200 reporting\_requirement

A **reporting\_requirement** is a type of **customer\_casting\_requirement** that is the information that shall be disclosed.

NOTE A **Reporting\_requirement** is defined according to 4.2.262.

EXPRESS specification:

\*)

```
ENTITY reporting_requirement
    SUBTYPE OF (customer_casting_requirement);
    WHERE
```

```
(* --- reporting_requirement to date (as date_to_be_reported)----- *)
```

```
wr1: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
    (ada.role.name='required delivery date')))=1);
```

```
(* reporting_requirement to customer_casting_requirement (as based_on)- *)

wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
(('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT'
IN TYPEOF (pdr.related_property_definition)
)) )) =1);

END_ENTITY; -- reporting_requirement
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'required delivery date' through **role**, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR2:** Exactly one reference that is the **relating\_action\_property** in exactly one **action\_property\_relationship** in which the **related\_action\_property** is a **customer\_casting\_requirement**.

**5.2.3.1.201 request\_for\_quotation**

A **request\_for\_quotation** is a type of **versioned\_action\_request** that is a solicitation from a customer to a casting or tooling supplier to bid or quote on producing cast parts or tooling as per a design specification.

NOTE A **Request\_for\_quotation** is defined according to 4.2.264.

EXPRESS specification:

```
*)
ENTITY request_for_quotation
  SUBTYPE OF (versioned_action_request);
  WHERE

  (* ----- quantity_breaks -----*)

wr1: SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
(((('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity_breaks'))
))>=1)) ))=0)) ))=0)) ))=0;

```

```

(* ----- quantity_per_release -----*)

wr2: SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity_per_release'))
)) <=1)) ))=0)) ))=0)) ))=0;

(* ----- quantity_per_month -----*)

wr3: SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity_per_month'))
)) <=1)) ))=0)) ))=0)) ))=0;

(* ----- quantity_per_year -----*)

wr4: SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity_per_year'))
)) <=1)) ))=0)) ))=0)) ))=0;

```

```

(* ----- quantity_per_order -----*)

wr5: SIZEOF(QUERY(ars <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'total quantity per order'))
))=1)) ))=0)) ))=0)) ))=0;

(* ----- type_of_quotation -----*)

wr6: SIZEOF(QUERY(ars <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'type of quotation'))
))=1)) ))=0)) ))=0)) ))=0;

(* - request_for_quotation to date (as delivery_date) - *)

wr7: (SIZEOF(QUERY(ada <* USEDIN(SELf ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='delivery date')))=1);

(* -- request_for_quotation to date (as reply_due_date) - *)

wr8: (SIZEOF(QUERY(ada <* USEDIN(SELf ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='reply due date')))=1);

(* -- request_for_quotation to date (as request_date) -*)

wr9: (SIZEOF(QUERY(ada <* USEDIN(SELf ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='request date')))=1);

```

ISO 10303-223:2008 (E)

```
(* -- request_for_quotation to date (as order_completion) - *)  
  
wr10: (SIZEOF(QUERY(ada <* USEDIN(SELF ,  
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |  
    (ada.role.name='order completion')))=1);  
  
    END_ENTITY; -- request_for_quotation  
(*
```

Formal propositions:

**WR1:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'quantity breaks'.

**WR2:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'quantity per release'.

**WR3:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'quantity per month'.

**WR4:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'quantity per year'.

**WR5:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'total quantity per order'.

**WR6:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_-representation** in which the **representation** is a **representation** in which exactly one **items** is a **descriptive\_representation\_item** with a **name** of 'type of quotation'.



**WR7:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'delivery date' through role, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR8:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'reply due date' through role, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR9:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'request date' through role, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**WR10:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'order completion' through role, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

### 5.2.3.1.202 reset\_or\_push\_pin

A **reset\_or\_push\_pin** is a type of **casting\_feature\_definition** that are the pins that reset the ejector system to its original position.

NOTE A **Reset\_or\_push\_pin** is defined according to 4.2.266.

#### EXPRESS specification:

\*)

```
ENTITY reset_or_push_pin
  SUBTYPE OF (ejector_design_feature);
  WHERE
```

```
(* ***** spring_description ***** *)
```

```
WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'spring_description')))) <=1 ))))=1;
```

ISO 10303-223:2008 (E)

```
(* - reset_or_return_pins to item_size (as pin_dimensions) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='reset or push pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'reset or push pin dimensions size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

END_ENTITY; -- reset_or_push_pin
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'spring description'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'reset or push pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'reset or push pin dimensions size usage' in which the **relying\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.203 resource\_with\_material

The **resource\_with\_material** is a type of **generic\_manufacturing\_resource** that defines an additional information for **material**.

NOTE 1 A **Resource\_with\_material** is defined according to 4.2.268.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.72 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY resource_with_material
    SUBTYPE OF (generic_manufacturing_resource);
WHERE

    wr1: (NOT (SIZEOF(QUERY( am <* SELF.usage |
        (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
        (NOT(SIZEOF(QUERY ( ama <* USEDIN(am,
            'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
            (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
            IN TYPEOF(ama)) AND
            (SIZEOF(QUERY ( eds <* ama.items |
                ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(eds))
                ))=1))))=0))))=0));
END_ENTITY; -- resource_with_material
(*

```

Formal propositions:

**WR1:** There shall exactly one references from a **action\_resource** of kind **action\_resource\_requirement** through the **resources** attribute that is referenced by a **product\_property\_association** through the **process** that references through **property\_or\_shape** a **product\_definition** with a **description** of 'resource material', which references a **product\_definition** through **definition**.

**5.2.3.1.204 resource\_with\_representation**

The **resource\_with\_representation** is a type of **generic\_manufacturing\_resource** that defines an additional information for external shape representation.

NOTE 1 A **Resource\_with\_representation** is defined according to 4.2.269.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.73 of ISO 10303-240:2005.

EXPRESS specification:

```

*)
ENTITY resource_with_representation
    SUBTYPE OF (generic_manufacturing_resource);
WHERE

    wr1:(NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
        (SIZEOF(QUERY ( rpr <* USEDIN(ap,
            'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
            ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS'
            IN TYPEOF(rpr.representation)) )) = 1) )) = 0));
END_ENTITY; -- resource_with_representation
(*

```

Formal propositions:

**WR1:** There shall be exactly one references from a **resource\_property** through the **resource** attribute that is referenced by a **resource\_property\_representation** through the **property** that references through **representation** exactly one **externally\_defined\_representation\_with\_parameters**.

### 5.2.3.1.205 revision

The **revision** is a type of **action\_relationship** that relates together different versions of a process plan.

NOTE 1 A **Revision** is defined according to 4.2.270.

NOTE 2 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.85 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
    ENTITY revision
SUBTYPE OF (ACTION_RELATIONSHIP);
WHERE

(* ----- reason_for_revision ----- *)

wr1: SIZEOF(QUERY ( rc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'RELATIONSHIP_CONDITION.APPLICABLE_RELATIONSHIPS') |
    (rc.name = 'reason for revision')) = 1;

(* ----- revision_level ----- *)

wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF.related_action.chosen_method,
    'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
    (('CAST_PARTS_SCHEMA.VERSIONED_ACTION_REQUEST'
    IN TYPEOF(ars.request)) AND
    (ars.request.description = 'revision level')))) = 1;

(* ----- revision to status_authority ( as approved_by) ----- *)

wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |
    ('CAST_PARTS_SCHEMA.APPROVAL'
    IN TYPEOF(adr.assigned_approval)))) >= 1;
```

```

(* ----- revision to process_plan_version (as related_to ----- *)
(* ----- revision to process_plan_version (as relating_to ----- *)

wr4: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
      IN TYPEOF(SELF.related_action)) AND
      ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
      IN TYPEOF(SELF.relatng_action));

END_ENTITY; -- revision
(*)

```

#### Formal propositions:

**WR1:** There shall be exactly one reference to **relationship\_condition** through **applicable\_relationships** with a name of 'reason for revision'.

**WR2:** There shall be exactly one **action** referenced through **relating\_action** that is an **action\_method** referenced by an **action\_request\_solution** with a **description** of 'revision level' through **method** that references a **versioned\_action\_request** through **request**.

**WR3:** The **revision** shall be referenced by exactly one **approval\_assignment** of kind **applied\_approval\_assignment** through the **items**.

**WR4:** There shall be a references a **process\_plan\_version** through **related\_action** and **relating\_relationship**.

### 5.2.3.1.206 rib

A **rib** is a type of **feature\_definition** that is a projecting wall feature item that strengthens a specific area of a casting.

NOTE A **Rib** is defined according to 4.2.271.

#### EXPRESS specification:

```

*)
ENTITY rib
    SUBTYPE OF (casting_design_feature);
WHERE

    wr1 : SELF\characterized_object.description IN ['planar','complex'];

```

ISO 10303-223:2008 (E)

```
(* ----- general_rib to numeric_parameter (as part_rib_fillet)----- *)

wr2: ((NOT (SELF\characterized_object.description = 'complex')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'part_rib_fillet')) ) =1 )))) =1 ));

(* ----- general_rib to numeric_parameter (as wall_to_floor_fillet)- *)

wr3: ((NOT (SELF\characterized_object.description = 'complex')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'wall to floor fillet')) ) =1 )))) =1 ));

(* ----- general_rib to numeric_parameter (as wall_to_top_fillet)----- *)

wr4: ((NOT (SELF\characterized_object.description = 'complex')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'wall to top radius')) ) =1 )))) =1 ));

(* ---- general_rib to face_shape_element(as walls_top_floor)----- *)
(* --- general_rib to planar_element(as walls_top_floor) ----- *)

WR5: ((NOT (SELF\characterized_object.description = 'complex')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION']
* TYPEOF(pdr.used_representation)) = 1)AND
(pdr.used_representation.name='walls top floor'))))>=1));
```

```
(* -----planar_rib to numeric_parameter (as part_rib_radius)--- *)

wr6: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'part_rib_radius')) )) =1 )))) =1 ));

(* --Planar_rib to numeric_parameter (as rib_extent_fillet)- *)

wr7: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'rib_extent_fillet')) )) =1 )))) =1 ));

(* - Planar_rib to numeric_parameter (as transition_to_top_radius)----- *)

wr8: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'transition to top radius')) )) =1 )))) =1 ));

(* - Planar_rib to numeric_parameter (as wall_to_top_radius) *)

wr9: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'wall to top radius')) )) =1 )))) =1 ));
```

ISO 10303-223:2008 (E)

```
(* - Planar_rib to numeric_parameter (as wall_to_floor_fillet)----- *)

wr10: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'wall to floor fillet')) ) =1 )))) =1 ));

(* ----- Planar_rib to face_shape_element(as top)----- *)
(* ----- Planar_rib to planar_element(as top) ----- *)
wr11: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION']
* TYPEOF(pdr.used_representation)) = 1)AND
(pdr.used_representation.name='top'))))>=1));

(* ----- Planar_rib to planar_element(as walls) ----- *)

WR12: ((NOT (SELF\characterized_object.description = 'planar')) OR
(SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION']
* TYPEOF(pdr.used_representation)) = 1)AND
(pdr.used_representation.name='walls'))))>=1));
END_ENTITY; -- rib
(*
```

Formal propositions:

**WR1:** The **description** shall be either 'planar' or 'complex'.

**WR2:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'part rib fillet'.

**WR3:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'wall to floor fillet'.

**WR4:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'wall to top radius'.

**WR5:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **face\_shape\_representation** or **planar\_shape\_representation** with a **name** of 'wall top floor'.



**WR6:** If the **description** is 'planar', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'part rib radius'.

**WR7:** If the **description** is 'planar', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'rib extent fillet'.

**WR8:** If the **description** is 'planar', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'transition to top radius'.

**WR9:** If the **description** is 'planar', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'wall to top radius'.

**WR10:** If the **description** is 'planar', the implicit representation shall contain exactly one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'wall to floor radius'.

**WR11:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **face\_shape\_representation** or **planar\_shape\_representation** with a **name** of 'top'.

**WR12:** If the **description** is 'complex', the implicit representation shall contain exactly one **representation\_item** of type **planar\_shape\_representation** with a **name** of 'wall'.

### 5.2.3.1.207 riser

A **riser** is a type of **casting\_feature\_definition** that is a reservoir of molten metal to be feed back into the casting as solidification occurs to prevent shrinkage.

NOTE A **Riser** is defined according to 4.2.273.

#### EXPRESS specification:

\*)

```
ENTITY riser
  SUBTYPE OF (gating_design_feature);
  WHERE
```

ISO 10303-223:2008 (E)

```
(* ***** riser_type ***** *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'riser type') AND
    ((it.description IN ['open', 'blind', 'open with radiation shield',
    'top', 'top_with_radiation_sheld'] )))))=1 ))))>=1;

(* ***** riser_sleeve ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'riser sleeve') AND
    ((it.description IN ['insulating', 'exothermic',
    'insulating and exothermic'] )))))<=1 ))))>=1;

(* ----- risers to item_size (as riser_dimensions) ----- *)

WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

(* ----- risers to item_size (as riser_neck) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser neck size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) >=0 )))) >=1;
```

```
(* ----- risers to riser_contact (as contact_definition) ----- *)

WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser contact reference usage') AND
  ('CAST_PARTS_SCHEMA.RISER_CONTACT'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* ----- risers to well (as riser_well) ----- *)

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'well reference usage') AND
  ('CAST_PARTS_SCHEMA.WELL' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* ----- risers to item_size (as riser_core) ----- *)

WR7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser core size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) <=1 )))) >=1;
  END_ENTITY; -- riser
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'riser type' and a **description** of either 'open', 'blind' or 'thermo'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'riser type' and a **description** of either 'insulating', 'exothermic' or 'insulating and exothermic'.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'riser occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'riser size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'riser occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'well reference usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'riser occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'riser contact reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **riser\_contact**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'riser occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'well reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **well**.

### 5.2.3.1.208 riser\_contact

A **riser\_contact** is a type of **casting\_feature\_definition** that is the shape and size of the feeder from the riser that attaches to the pattern.

NOTE A **Riser\_contact** is defined according to 4.2.272.

#### EXPRESS specification:

\*)

```
ENTITY riser_contact
  SUBTYPE OF (gating_design_feature);
  WHERE
```

```
(* ----- contact_placement ----- *)
```

```
WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
  ((it.description IN ['cope', 'drag', 'cope and drag']
  )) )) )) =1 )) ))>=1;
```

```
(* ----- riser_contact to shape_element (as break_off_connection)----- *)
wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='riser contact occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser contact shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) <=0 ))) <=1;

END_ENTITY; -- riser_contact
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'cope or drag type' and a **description** of either 'cope', 'drag' or 'cope and drag'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'riser contact occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'riser contact shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

### 5.2.3.1.209 runner

A **runner** is a type of **casting\_feature\_definition** that is a channel that connects the sprue to each ingate through which the molten metal flows into the mould cavity.

NOTE A **Runner** is defined according to 4.2.274.

### EXPRESS specification:

```
*)
  ENTITY runner
  SUBTYPE OF (gating_design_feature);
  WHERE

(* ----- runner_placement ----- *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
  ((it.description IN ['cope', 'drag', 'cope and drag'] ))
  )) ) ) =1 ) ) )>=1;
```

ISO 10303-223:2008 (E)

(\* ----- runners to feeders (as riser\_connection) ----- \*)

```
WR2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'feeder reference usage') AND
  ('CAST_PARTS_SCHEMA.FEEDER' IN TYPEOF(sar.relying_shape_aspect))
  )) >=0 )))) >=1;
```

(\* ----- runners to item\_size (as runner\_dimensions) ----- \*)

```
WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'runner dimensions size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;
```

(\* ----- runners to sprue ( as sprue\_connections) ----- \*)

```
WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue reference usage') AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;
```

(\* ----- runners to filter ( as metal\_flow\_filter) ----- \*)

```
WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'filter reference usage') AND
  ('CAST_PARTS_SCHEMA.FILTER' IN TYPEOF(sar.relying_shape_aspect))
  )) >=0 )))) >=1;
```

```
(* ----- runners to choke ( as runner choke) ----- *)

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'choke reference usage') AND
  ('CAST_PARTS_SCHEMA.CHOKE' IN TYPEOF(sar.relater_shape_aspect))
  )) >=0 )))) >=1;

(* ----- runners to vent ( as runner vent) ----- *)

WR7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'vent reference usage') AND
  ('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relater_shape_aspect))
  )) >=0 )))) >=1;

END_ENTITY; -- runner
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'runner placement' and a **description** of either 'cope', 'drag' or 'cope and drag'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'feeder reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **feeder**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'runner dimensions size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'sprue reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **sprue**.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'filter reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **filter**.

**WR6:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'choke reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **choke**.

**WR7:** Exactly one **shape\_aspect** with a **description** of 'runner occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'vent reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **vent**.

### 5.2.3.1.210 sampled\_set

A **sampled\_set** is a type of **property\_definition** that is a group of parts that are inspected or tested together.

NOTE A **Sampled\_set** is defined according to 4.2.275.

#### EXPRESS specification:

\*)

```
ENTITY sampled_set
  SUBTYPE OF (property_definition);
  WHERE
```

```
(* ----- size----- *)
```

```
wr1:(NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY( it <* pdr.used_representation.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
  (it.name = 'size') )) <=1)) ) =0));
```



```

(* ----- sampled_set to master_sample(as master_sample_inspected)-- *)

(* **** SEE MASTER_SAMPLE FOR THIS RULE **** *)

(* ----- sampled_set to heat(as sampled_form)----- *)
(* ----- sampled_set to lot(as sampled_form)----- *)

Wr2: (NOT(SIZEOF(QUERY ( da <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS') |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.LOT_EFFECTIVITY',
    'CAST_PARTS_SCHEMA.DATED_EFFECTIVITY'] *
    TYPEOF(da\effectivity_assignment.assigned_effectivity)) = 1)))=0));

(* -- to product_quality_report (as quality_assurance_document)- *)

Wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'production quality report'))
  >= 1;

  END_ENTITY; -- sampled_set
(*

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_-representation\_item** and **length\_measure\_with\_unit** with a **name** of 'size'.

**WR2:** Exactly one reference that is the **items** in exactly one **applied\_effectivity\_assignment** of type **effectivity\_assignment** in which the **assigned\_effectivity** is an **lot\_effectivity** or a **dated\_effectivity**

**WR3:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** in which the **kind** is exactly one **document\_kind** with **product\_data\_type** of 'production quality report'.

### 5.2.3.1.211 sand\_cast\_design\_feature

The **sand\_cast\_design\_feature** is a type of **casting\_feature\_definition**, that defines sand casting features on a cast part.

NOTE A **Sand\_cast\_design\_feature** is defined according to 4.2.276.

#### EXPRESS specification:

```

*)
ENTITY sand_cast_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE

```

ISO 10303-223:2008 (E)

```
WR1: SIZEOF(QUERY( pdr <*
  get_property_definition_representations (SELF) |
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF ( pdr.used_representation ) ) ) =1;

(* ***** Sand_cast_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 ))))=1;

END_ENTITY; -- sand_cast_design_feature
(*
```

Formal propositions:

**WR1:** The **feature\_definition** shall have exactly one implicit **representation** which shall be of type **shape\_representation\_with\_parameters**.

**WR2:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **placement** through the **used\_representation** that has a **name** of 'orientation'.

### 5.2.3.1.212 sand\_casting\_tooling

A **sand\_casting\_tooling** is a type of **casting\_product\_definition** that defines the resources necessary to produce a cast part.

NOTE A **Sand\_casting\_tooling** is defined according to 4.2.277.

EXPRESS specification:

```
*)
ENTITY sand_casting_tooling
  SUBTYPE OF (casting_product_definition);
  WHERE

(* ----- sand_casting_tooling to material (as material_definition) *)

WR1: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='material_definition') AND
  (SIZEOF(QUERY (mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
  ))=1))))=0));
```

```

(* ----- sand_casting_tooling to material (as tooling_material) *)

WR2: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='tooling material') AND
  (SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (mfuo.name IN ['wood','cast iron','urethane','aluminum'] )
  ))=1))))=0));

(* to production_pattern_definition (as pattern_equipment) *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sand casting tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production pattern definition reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION'
  IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;

END_ENTITY; -- sand_casting_tooling
(*

```

#### Formal propositions:

**WR1:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'material definition' and of type **make\_from\_usage\_option** in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR2:** Exactly one reference that is the **relating\_product\_definition** in exactly one **product\_definition\_relationship** with **description** of 'tooling material' and of type **make\_from\_usage\_option** with name of 'wood','cast iron','urethane','aluminum' in which the **related\_product\_definition** is a **product\_definition** that has one reference that is the **definition** in exactly one **material\_designation**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'sand casting tooling occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'production pattern definition reference usage' and of type **shape\_defining\_relationship** in which the **relaying\_shape\_aspect** is a **production\_pattern\_definition**.

### 5.2.3.1.213 sand\_mould

A **sand\_mould** is a type of **casting\_product\_definition** that is a type of casting into which liquid metal is poured and allowed to solidify to form a cast part.

NOTE A **Sand\_mould** is defined according to 4.2.278.

EXPRESS specification:

```

*)
ENTITY sand_mould
  SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- sand_mould_type ----- *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'sand mould type') AND
  ((it.description IN ['grean sand material','skin dried metrhold',
  'cold box mould method','no bake mould method'] ) ) ) ) =1 ) ) )>=1;

(* ***** sand_mould to chaplet (as chaplet_definition) ***** *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sand mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chaplet reference usage') AND
  ('CAST_PARTS_SCHEMA.CHAPLET' IN TYPEOF(sar.relater_shape_aspect))
  )) >=0 )))) >=1;

(* ***** sand_mould to chill (as chill_definition) ***** *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sand mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chill reference usage') AND
  ('CAST_PARTS_SCHEMA.CHILL' IN TYPEOF(sar.relater_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- sand_mould to equipment (as mould_made_by) *)

wr4: (NOT (SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
(('CAST_PARTS_SCHEMA.ACTION_METHOD'
IN TYPEOF(adr.assigned_action_method)) AND
(SIZEOF(QUERY(ar <* USEDIN(adr.assigned_action_method,
'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(ar) )))=1))))=0));

END_ENTITY; -- sand_mould
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'sand mod characteristics' and a **description** of either 'green sand material', 'skin dried method', 'cold box mould method' or 'no bake mould method'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'sand mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chaplet reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chaplet**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'sand mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_aspect\_relationship** in with a **description** of 'chill reference usage' and of type **shape\_defining\_relationship** in which the **relating\_shape\_aspect** is a **chill**.

**WR4:** Exactly one reference that is the **items** in exactly one **applied\_action\_method\_assignment** of type **action\_method\_assignment** which the **assigned\_action\_method** was an **action\_method** that has one reference that is the **usage** in exactly one **action\_resource** of type **casting\_equipment**.

### 5.2.3.1.214 secondary\_tooling

A **secondary\_tooling** is a type of **casting\_feature\_definition** that is a supplementary devices used to facilitate the casting process.

NOTE A **Secondary\_tooling** is defined according to 4.2.279.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY secondary_tooling
  SUBTYPE OF (die_design_feature);
WHERE

(* ***** type_of_tooling ***** *)

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'type of tooling')))) =1 )))=1;
END_ENTITY; -- secondary_tooling
(*
```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'type of tooling'.

### 5.2.3.1.215 shakeout

The shakeout is a type of casting\_equipment that is the stage in the casting process where the sand from the mould is cleaned off of the newly formed castings through vigorous vibration.

NOTE A **Shakeout** is defined according to 4.2.281.

EXPRESS specification:

```
*)

ENTITY shakeout
  SUBTYPE OF (casting_equipment);
END_ENTITY;
(*
```

### 5.2.3.1.216 shape\_dimension

A **shape\_dimension** is a type of **property\_definition** that defines the modified shape dimensions that are required to produce a cast part and still satisfy the form, fit and function specified by the customer.

NOTE A **Shape\_dimension** is defined according to 4.2.284.

EXPRESS specification:

```

*)
  ENTITY shape_dimension
    SUBTYPE OF (property_definition);
    WHERE

(* --shape_dimension to dimensional_tolerance(as dimensionals)----- *)

(* -- shape_dimension to production_die_mould (as tooling)----- *)
(* -- shape_dimension to production_core_box (as tooling)----- *)
(* -- shape_dimension to production_pattern_definition (as tooling)---*)
(* -- shape_dimension to production_investment_cast_mould (as tooling) *)
(* -- shape_dimension to production_die_cast_mould (as tooling)----- *)

wr1: (NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  (('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE'
  IN TYPEOF(pdr.related_property_definition)) AND
  (NOT (SIZEOF(QUERY( it <* USEDIN(pdr.related_property_definition,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  (SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD',
  'CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX',
  'CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION',
  'CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD',
  'CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD']
  * TYPEOF(it)) = 1)
  )) =1 )) )) =0 ));

  END_ENTITY;
(*

```

Formal propositions:

**WR1:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_definition\_relationship** in which the **relating\_property\_definition** is an **shape\_aspect** that has exactly one reference through the **of\_shape** that is a **production\_die\_mould**, **production\_pattern\_definition**, **production\_core\_box**, **production\_die\_cast\_mould** or **production\_investment\_cast\_mould**.

**5.2.3.1.217 shot\_sleeve**

A **shot\_sleeve** is a type of **casting\_feature\_definition** that is the conduit from the diecasting machine that provides molten metal to die mould cavity.

NOTE A **Shot\_sleeve** is defined according to 4.2.288.

EXPRESS specification:

```

*)
    ENTITY shot_sleeve
    SUBTYPE OF (die_design_feature);
    WHERE

    (* ----- shot_sleeve to item_size (as shot_sleeve_size) ----- *)

    WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='shot sleeve occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'shot sleeve shape usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
        IN TYPEOF(sar.relying_shape_aspect))
        )) =1 ))) >=1;
    END_ENTITY; -- shot_sleeve
    (*

```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'shot sleeve occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'shot sleeve shape usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.218 simulated\_property

A **simulated\_property** is a type of **representation** that is a physical variable whose value is predicted by a simulation.

NOTE A **Simulated\_property** is defined according to 4.2.289.

EXPRESS specification:

```

*)

ENTITY simulated_property
    SUBTYPE OF (representation);
    WHERE

```



```
(*    property_value    *)
```

```
WR1: SIZEOF(QUERY( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.PARAMETER_VALUE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'property_value'))))=1;
```

```
END_ENTITY; -- simulated_property
(*
```

### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **parameter\_value** with a **name** of 'property value'.

### **5.2.3.1.219 simulation\_input**

A **simulation\_input** is a type of **requirement\_for\_action\_resource** that is the collection of data required by simulation software in order to begin a **simulation\_run**.

NOTE A **Simulation\_input** is defined according to 4.2.291.

### EXPRESS specification:

```
*)
```

```
ENTITY simulation_input
  SUBTYPE OF (requirement_for_action_resource);
  WHERE
```

```
(* simulation_input to CAST_PARTS_SCHEMA (as setup_file)    *)
```

```
wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
  (adr.assigned_document))))=1;
```

```
(* simulation_input to condition_or_assumption(as condition) *)
```

```
wr2: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.CONDITION_OR_ASSUMPTION' IN TYPEOF(ap) ) AND
  (ap.name='condition'))
  )) >= 0;
```

```
(* simulation_input to condition_or_assumption(as assumption) *)
```

```
wr3: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.CONDITION_OR_ASSUMPTION' IN TYPEOF(ap) ) AND
  (ap.name='assumption'))
  )) >= 0;
```

ISO 10303-223:2008 (E)

```
(* to integration_interval(as integration_time_specification) *)

wr4: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.INTEGRATION_INTERVAL' IN TYPEOF(ap) ) AND
  (ap.name='assumption'))
  )) >= 1;

(* simulation_input to numeric_parameter(as input_parameter) *)

wr5: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE' ) |
  (NOT(SIZEOF(QUERY ( ap <* USEDIN(rp,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY' ) |
  ((SIZEOF(QUERY(ri <* ap.representation.items|
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(ri)) AND
  (ri.name = 'input parameter')))) >=1 )))) =0 )))) =0 ));

(* simulation_input to property_relationship(as input_parameter) *)

wr6: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE' ) |
  (SIZEOF(QUERY ( ap <* USEDIN(rp,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY' ) |
  (('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN
  TYPEOF(ap.representation) ) )) >=1 )))) =0));

(* simulation_input to simulation_input_region (as region) *)

wr7: SIZEOF(QUERY ( r <* SELF.resources |
  'CAST_PARTS_SCHEMA.SIMULATION_INPUT_REGION'
  IN TYPEOF(r))) >= 1;

END_ENTITY; -- simulation_input
(*
```

Formal propositions:

**WR1:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document\_reference**.

**WR2:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **condition\_or\_assumption** with **name** of 'condition'.

**WR3:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **condition\_or\_assumption** with **name** of 'assumption'.

**WR4:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **integration\_interval** with **name** of 'assumption'.

**WR5:** Exactly one reference that is the **resource** in exactly one **resource\_property** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** awith a **name** of 'input parameter'.

**WR6:** Exactly one reference that is the **resource** in exactly one **resource\_property** in which the **representation** is a **representation** in which exactly one **items** is a **property\_relationship** awith a **name** of 'input parameter'.

**WR7:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **simulation\_input\_region**.

### 5.2.3.1.220 simulation\_input\_region

A **simulation\_input\_region** is a type of **action\_resource** that is a portion of space containing similar material properties at the beginning of a **simulation\_run**.

NOTE A **Simulation\_input\_region** is defined according to 4.2.292.

#### EXPRESS specification:

```
*)
ENTITY simulation_input_region
  SUBTYPE OF (action_resource);
  WHERE

(* simulation_input_region to boundary_conditon(as boundary_condition) *)

wr1: SIZEOF(QUERY( am <* SELF.usage |
  ('CAST_PARTS_SCHEMA.BOUNDARY_CONDITION' IN TYPEOF(am))))>=0;

(* simulation_input_region to boundary_conditon(as boundary_condition) *)

wr2: SIZEOF(QUERY( am <* SELF.usage |
  ('CAST_PARTS_SCHEMA.MESHING_CONDITION' IN TYPEOF(am))))>=0;
```

(\* simulation\_input\_region to numeric\_parameter(as region\_property)\*)

```

wr3: (NOT (SIZEOF(QUERY( am <* SELF.usage |
(('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
(NOT(SIZEOF(QUERY( ama <* USEDIN(am,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(NOT (SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
(NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(mri <* pd.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(mri)) = 2)AND
(mri.name = 'region property'))
)) >=1 ))) =0 ))))) =0 ))))) =0 ))))) =0 ))));

```

(\* simulation\_input\_region to property\_relationship(as region\_property) \*)

```

wr4: (NOT (SIZEOF(QUERY( am <* SELF.usage |
(('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
(NOT(SIZEOF(QUERY( ama <* USEDIN(am,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(NOT (SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
(NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(mri <* pd.used_representation.items |
('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN TYPEOF(mri))))=1)
)) >=1 ))))) =0 ))))) =0 ))))) =0 ))));

```

(\* simulation\_input\_region to shape\_aspect (as area\_to) \*)

```

wr5: (NOT (SIZEOF(QUERY( am <* SELF.usage |
(('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
(NOT(SIZEOF(QUERY( ama <* USEDIN(am,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama))
AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
)) =1 ))))) =0 ))))) =0 ))));

```

```
(* simulation_input_region to part_version (as associated_part) *)

wr6: (NOT (SIZEOF(QUERY( am <* SELF.usage |
  (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
  (NOT(SIZEOF(QUERY( ama <* USEDIN(am,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (eds))
  )) >=0 )))) =0 )))) =0));

END_ENTITY; -- simulation_input_region
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **boundary\_condition**.

**WR2:** Exactly one reference that is the **resource** in exactly one **resource\_property** of type **meshing\_condition**.

**WR3:** Exactly one reference that is the **resource** in exactly one **resource\_property** in which the **representation** is a **representation** in which exactly one **items** is a **measure\_representation\_item** awith a **name** of 'region property'.

**WR4:** Exactly one reference that is the **resource** in exactly one **resource\_property** in which the **representation** is a **representation** in which exactly one **items** is a **property\_relationship** awith a **name** of 'region property'.

**WR5:** Exactly one reference of **usage** is a **action** of type **property\_process** that has exactly one reference that is the **process** in exactly one **process\_property\_association** that has exactly one reference of **property\_or\_shape** in exactly one **shape\_aspect**.

**WR6:** Exactly one reference of **usage** is a **action** of type **product\_definition\_process** that has exactly one reference that is the **process** in exactly one **process\_product\_association** that has exactly one reference of **defined\_product** in exactly one **product\_definition**.

### 5.2.3.1.221 simulation\_output

A **simulation\_output** is a type of **versioned\_action\_request** that is the data produced by a simulation software as the result of a **simulation\_run**.

NOTE A **Simulation\_output** is defined according to 4.2.293.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
ENTITY simulation_output
  SUBTYPE OF (versioned_action_request);
  WHERE

(* simulation_output to defect_prediction(as predicted_defect) *)

wr1: SIZEOF(QUERY ( ars <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') | (
  ('CAST_PARTS_SCHEMA.DEFECT_PREDICTION' IN TYPEOF(ars.method))))>=0;

(* simulation_output to simulation_result(as result) *)

wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') | (
  'CAST_PARTS_SCHEMA.SIMULATION_RESULT' IN TYPEOF(ars)) )) >= 0;

END_ENTITY; -- simulation_output
(*
```

Formal propositions:

**WR1:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** in which the **method** is an **action\_method** that has exactly one reference that is the **definition** in exactly one **action\_property** that has exactly one reference that is the **property** in exactly one **action\_property\_representation** of type **defect\_prediction**.

**WR2:** Exactly one reference that is the **request** in exactly one **action\_request\_solution** of type **simulation\_result**.

### 5.2.3.1.222 simulation\_output\_region

A **simulation\_output\_region** is a type of **action\_property** that is a portion of space containing similar material properties at the end of a **simulation\_run**.

NOTE A **Simulation\_output\_region** is defined according to 4.2.294.

EXPRESS specification:

```

*)

ENTITY simulation_output_region
  SUBTYPE OF (action_property);
WHERE

(* simulation_output_region to simulation_input_region(as belongs_to) *)

wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(SELF.definition)) AND
      (SIZEOF(QUERY ( am <* USEDIN(SELF.definition,
        'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') | (
        'CAST_PARTS_SCHEMA.SIMULATION_INPUT_REGION' IN TYPEOF(am)) ))
      = 1);

(* simulation_output_region to simulation_unit_state(as history) *)

wr2: SIZEOF(QUERY ( am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
  ('CAST_PARTS_SCHEMA.SIMULATION_UNIT_STATE'
  IN TYPEOF(am.related_action_property)
  )))>=1;

(* simulation_output_region to shape_element(as region_shape) *)

wr3: ('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(SELF.definition)) AND
      (NOT(SIZEOF(QUERY(am <* USEDIN(SELF.definition,
        'CAST_PARTS_SCHEMA.'+
        'ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
        IN TYPEOF(am)) AND
        (SIZEOF(QUERY(eds <* am.items |
        ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
        )) =1 ))))=0)));

END_ENTITY; -- simulation_output_region
(*

```

Formal propositions:

**WR1:** Exactly one reference of **definition** is a **action\_method** that has exactly one reference that is the **usage** is an **action\_resource** of type **simulation\_input\_region**.

**WR2:** Exactly one reference of **property** is a **action\_property\_representation** of type **simulation\_unit\_state**.

**WR3:** Exactly one reference of **definition** is a **property\_process** that has one reference that is the **process** in exactly one **process\_property\_association** in which the **property\_or\_shape** is a **shape\_aspect**.

### 5.2.3.1.223 simulation\_process

A **simulation\_process** is a type of **casting\_activity** that is the metalcasting process used by the foundry to be simulated by the simulation run.

NOTE A **Simulation\_process** is defined according to 4.2.295.

EXPRESS specification:

\*)

```

ENTITY simulation_process
  SUBTYPE OF (casting_activity);
  WHERE
    (* --- simulation_process to design_part (as part_geometry) ----- *)
    (* --- simulation_process to design_part (as part_geometry) ----- *)
    (* --- simulation_process to design_part (as cast_part_with_rigging) -- *)

wr1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (NOT(SIZEOF(QUERY ( ip <* sar.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.CAST_PART',
    'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
    'CAST_PARTS_SCHEMA.DESIGN_PART'] * TYPEOF(ip)) =1))
  ) =0 ))))>=0);

(* -- to customer_casting_requirement (as quality_requirements) -- *)

wr2: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT' IN TYPEOF (eds)
  ))) >=1 ))))=0));

(* -- simulation_process to simulation_run (as run_simulation) -- *)

wr3: (SIZEOF(QUERY(ar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD') |
  ('CAST_PARTS_SCHEMA.SIMULATION_RUN' IN TYPEOF(ar) )) )=1);

(* ---- to tooling(as tooling_for_simulation) ----- *)

wr5: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (eds)) AND
  (SIZEOF(['CAST_PARTS_SCHEMA.DIE_CAST_TOOLING',

```



```
'CAST_PARTS_SCHEMA.DIE_MOULD_TOOLING',
'CAST_PARTS_SCHEMA.SAND_CASTING_TOOLING',
'CAST_PARTS_SCHEMA.CORE_BOX_TOOLING',
'CAST_PARTS_SCHEMA.INVESTMENT_CASTING_TOOLING'] * TYPEOF(eds) =1)
))) =1 )))) =0));
```

```
END_ENTITY; -- simulation_process
(*
```

#### Formal propositions:

**WR1:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **product\_definition** of type **cast\_part**, **cast\_part\_with\_rigging**, or **design\_part**.

**WR2:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** of type **customer\_casting\_requirement**.

**WR3:** There shall be exactly one reference through the **chosen\_method** attribute to an **action\_method** of type **simulation\_run**.

**WR4:** There shall be exactly one reference through the **assigned\_action\_method** attribute of an **action\_method\_assignment** that is of type **applied\_action\_method\_assignment** that references through the **items** attribute a **property\_definition** of type **die\_cast\_tooling**, **die\_mould\_tooling**, **sand\_casting\_tooling**, **core\_box\_tooling** or **investment\_casting\_tooling**.

### 5.2.3.1.224 simulation\_result

A **simulation\_result** is a type of **action\_request\_solution** that is a specific outcome during a specific time period of a **simulation\_run**.

NOTE A **Simulation\_result** is defined according to 4.2.296.

#### EXPRESS specification:

\*)

```
ENTITY simulation_result
  SUBTYPE OF (action_request_solution);
  WHERE

  (* begin_time *)

  wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
    ((adr.role.name='begin_time') AND
    ('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF
    (adr.assigned_time))))))=1;
```

## ISO 10303-223:2008 (E)

```
(* end_time *)

wr2: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
  ((adr.role.name='end time') AND
  ('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF
  (adr.assigned_time))))=1;

(* simulation_result to simulation_output_region (as region) *)

wr3: (NOT(SIZEOF(QUERY( am <* USEDIN(SELF.method,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
  (SIZEOF(QUERY( sor <* USEDIN(am.related_method,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.SIMULATION_OUTPUT_REGION' IN TYPEOF(sor))
  )) >=1 ))=0));

(* imulation_result to illustration (as illustration) *)

wr4: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
  (adr.assigned_document))))>=0;

END_ENTITY; -- simulation_result
(*
```

### Formal propositions:

**WR1:** There shall exactly one references from a **applied\_time\_assignment** through the **items** of type **time\_assignment** that has one reference of **role** that is an **time\_role** with **name** of 'begin time' that references a **local\_time** through the **assigned\_time**.

**WR2:** There shall exactly one references from a **applied\_time\_assignment** through the **items** of type **time\_assignment** that has one reference of **role** that is an **time\_role** with **name** of 'end time' that references a **local\_time** through the **assigned\_time**.

**WR3:** Exactly one reference of **chosen\_method** is a **action\_method** of type **simulation\_output\_region**.

**WR4:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **illustration**.

### 5.2.3.1.225 simulation\_run

A **simulation\_run** is a type of **executed\_action** that is an execution of a simulation software program to model a casting process.

NOTE A **Simulation\_run** is defined according to 4.2.298.

EXPRESS specification:

```

*)

ENTITY simulation_run
  SUBTYPE OF (directed_action);
  WHERE

(* identification *)

wr1: SIZEOF(QUERY( arr <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ID_ATTRIBUTE.IDENTIFIED_ITEM') |
  ('CAST_PARTS_SCHEMA.ID_ATTRIBUTE'
  IN TYPEOF(arr))))=1;

(* simulation_run to activity (as associated_process) *)

wr2: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY'
  IN TYPEOF(SELF.chosen_method));

(* simulation_run to simulation_input (as input) *)

wr3: SIZEOF(QUERY( arr <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
  ('CAST_PARTS_SCHEMA.SIMULATION_INPUT'
  IN TYPEOF(arr))))=1;

(* simulation_run to simulation_output (as output) *)

wr4: SIZEOF(QUERY ( r <* SELF.directive.requests |
  'CAST_PARTS_SCHEMA.SIMULATION_OUTPUT'
  IN TYPEOF(r))) = 1;

(* simulation_run to simulation_software (as software_used) *)

wr5: SIZEOF(QUERY( arr <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('CAST_PARTS_SCHEMA.SIMULATION_SOFTWARE'
  IN TYPEOF(arr))))=1;

END_ENTITY; -- simulation_run
(*

```

Formal propositions:

**WR1:** Exactly one reference that is the **identified\_item** in exactly one **id\_attribute**.

**WR2:** Exactly one reference of **chosen\_method** is a **action\_method** of type **process\_plan\_activity**.

ISO 10303-223:2008 (E)

**WR3:** Exactly one reference by an **action\_resource\_requirement** of type **simulation\_input** through the **operations**.

**WR4:** Exactly one reference of **directive** is a **directive\_action** which the **request** is a **simulation\_output**.

**WR5:** Exactly one reference that is the **usage** in exactly one **action\_resource** of type **simulation\_software**.

### 5.2.3.1.226 simulation\_run\_relationship

A **simulation\_run\_relationship** is a type of **action\_method\_relationship** that is an association between two different **simulation\_run**.

NOTE A **Simulation\_run\_relationship** is defined according to 4.2.299.

EXPRESS specification:

\*)

```
ENTITY simulation_run_relationship
  SUBTYPE OF (action_method_relationship);
  WHERE
```

```
(* simulation_run_relationship to simulation_run (as related) *)
```

```
wr1: ('CAST_PARTS_SCHEMA.SIMULATION_RUN'
      IN TYPEOF(SELF.related_method));
```

```
(* simulation_run_relationship to simulation_run (as relating) *)
```

```
wr2: ('CAST_PARTS_SCHEMA.SIMULATION_RUN'
      IN TYPEOF(SELF.relying_method));
```

```
END_ENTITY;
```

```
(*
```

Formal propositions:

**WR1:** Exactly one reference of **related\_method** is a **simulation\_run**.

**WR2:** Exactly one reference of **relying\_method** is a **simulation\_run**.

### 5.2.3.1.227 simulation\_software

A **simulation\_software** is a type of **action\_resource** that is the computer program used to model the casting process.

NOTE A **Simulation\_software** is defined according to 4.2.300.

EXPRESS specification:

```

*)

ENTITY simulation_software
  SUBTYPE OF (action_resource);
  WHERE

  (* simulation_software to date (as release_date) *)

  wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ((adr.role.name='release date') AND
    ('CAST_PARTS_SCHEMA.DATE' IN TYPEOF
    (adr.assigned_date))))))=1;

END_ENTITY; -- simulation_software
(*)

```

Formal propositions:

**WR1:** There shall be exactly one reference by a **applied\_date\_assignment** through **items** that references **date\_role** with a **name** of 'release date' through role, and is of kind **date\_assignment** that references a **calendar\_date** through **assigned\_date**.

**5.2.3.1.228 simulation\_unit**

A **simulation\_unit** is a type of **action\_property\_representation** that is a point or region which is not broken down further for the purpose of simulation.

NOTE A **Simulation\_unit** is defined according to 4.2.302.

EXPRESS specification:

\*)

```
ENTITY simulation_unit
  SUBTYPE OF (action_property);
WHERE
```

(\* simulation\_unit to location\_element(as unit\_location) \*)

```
wr1:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF.definition,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama))
  AND
  (NOT (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
  (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(mri <* pd.used_representation.items |
  ('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
  IN TYPEOF(mri))))=1)
  ))=1))))=0))))=0));
```

(\* simulation\_unit to simulation\_output\_region(as belongs\_to) \*)

```
wr2: SIZEOF(QUERY ( am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
  ('CAST_PARTS_SCHEMA.SIMULATION_OUTPUT_REGION'
  IN TYPEOF(am.related_action_property)
  ))=1;
```

(\* simulation\_unit to simulation\_unit\_state(as history)\*)

```
wr4: SIZEOF(QUERY ( am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
  ('CAST_PARTS_SCHEMA.SIMULATION_UNIT_STATE'
  IN TYPEOF(am.related_action_property)
  ))>=1;
```

```
(* simulation_unit to shape_element(as unit_shape) *)

wr5: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF.definition,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
)) =1 )) )=0)));

END_ENTITY; -- simulation_unit
(*
```

### Formal propositions:

**WR1:** Exactly one **representation** used for the implicit representation shall be of type **location\_shape\_representation**.

**WR2:** Exactly one reference of **property** is a **action\_property** of type **simulation\_output\_region**.

**WR3:** Exactly one reference of **property** is a **action\_property** that has exactly one reference that is the **relating\_action\_property** in exactly one **action\_property\_relationship** in which the **related\_action\_property** is a **simulation\_unit\_state**.

**WR4:** Exactly one **representation** used for the implicit representation shall be of type **shape\_definition\_representation** that has exactly one reference of **of\_shape** is a **shape\_aspect**.

### 5.2.3.1.229 simulation\_unit\_state

A **simulation\_unit\_state** is a type of **action\_property\_representation** that is the condition of the material inside a **simulation\_unit** at a particular value of simulation time.

NOTE A **Simulation\_unit\_state** is defined according to 4.2.303.

### EXPRESS specification:

```
*)

ENTITY simulation_unit_state
  SUBTYPE OF (action_property);
  WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
    ((adr.role.name = 'evaluation_time') AND
    ('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF(adr.assigned_time)))
  )) = 1;
```

ISO 10303-223:2008 (E)

```
wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    'CAST_PARTS_SCHEMA.SIMULATED_PROPERTY'
    IN TYPEOF(apr.representation)
    ))>=1;
END_ENTITY; -- simulation_unit_state
(*
```

Formal propositions:

**WR1:** There shall exactly one references from a **applied\_time\_assignment** through the **items** of type **time\_assignment** that has one reference of **role** that is an **time\_role** with **name** of 'evaluation time' that references a **local\_time** through the **assigned\_time**.

**WR2:** Exactly one reference of **representation** is a **simulated\_property**.

### 5.2.3.1.230 single\_activity\_relationship

The **single\_activity\_relationship** is a type of **sequential\_method** that relates a **manufacturing\_process** to a type of setup.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.2.78 of ISO 10303-240:2005.

EXPRESS specification:

```
*)
ENTITY single_activity_relationship
  SUBTYPE OF (sequential_method);
  WHERE
    wr2: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
      relating_method);
    wr3: SIZEOF(['CAST_PARTS_SCHEMA.MACHINE_SETUP',
      'CAST_PARTS_SCHEMA.ANCILLARY_SETUP'] * TYPEOF(SELF.
      related_method)) = 1;
  END_ENTITY; -- single_activity_relationship
(*
```

Formal propositions:

**WR1:** The **relating\_method** attribute shall be **manufacturing\_process**.

**WR2:** The **related\_method** attribute shall be either a **machine\_setup** or **ancillary\_setup**



### 5.2.3.1.231 slide

A **slide** is a type of **casting\_feature\_definition** that is used in a die mould to form some portion of a cast part that can not be form and drawn by the die mould.

NOTE A **Slide** is defined according to 4.2.305.

#### EXPRESS specification:

```

*)
  ENTITY slide
  SUBTYPE OF (die_design_feature);
  WHERE

(* ***** slide_material ***** *)

wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN
  ['wood','cast iron','urethane','aluminum','plastic']))) ) = 1;

(* ----- slide to shape_element (as shape)----- *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='slide occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'slide shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

(* ***** slide_activation_method ***** *)

w3:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'slide activation method'))))
  =1 ))))=1;

  END_ENTITY; -- slide
(*)

```

Formal propositions:

**WR1:** Exactly one reference that is the **definitions** in exactly one **material\_designation** with **name** of 'soft wood', 'hard wood', 'urethane', 'epoxy', 'aluminum', 'iron', 'steel', 'wood material reinforced', 'other'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'slide occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'slide shape usage' in which the **relating\_shape\_aspect** is a **shape\_aspect**.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'slide activation method'.

### 5.2.3.1.232 special\_inspection\_requirement

A **special\_inspection\_requirement** is a type of **customer\_casting\_requirement** that is a specific method for evaluation and qualification of the cast part for acceptance by the customer.

NOTE A **Special\_inspection\_requirement** is defined according to 4.2.306.

EXPRESS specification:

\*)

```
ENTITY special_inspection_requirement
    SUBTYPE OF (customer_casting_requirement);
```

WHERE

(\* ----- inspection\_description ----- \*)

```
wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND
    (it.name = 'inspection description') AND
    (it.description IN (['ultrasonic', 'other testing method',
    'pressure test', 'x ray radiography',
    'welding', 'liquid penetrant',
    'magnetic particle'] ) )
    ) )) =1 )))=0));
```

(\* ----- other\_testing\_method ----- \*)

```
wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND
    (it.name = 'other testing method')
    ) ))<=1 )))) = 0);
```

```
(* ----- performed_by_customer ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
  AND
  (it.name = 'performed by customer') AND
  (it.description IN (['TRUE','FALSE'] ) )
  ) ) ) >=1 )))=0));

(* - to specification (as inspection_requirements)----- *)

wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'inspection requirements') ) )
  >= 1;

END_ENTITY; -- special_inspection_requirement
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'inspection description' and a **description** of either 'ultrasonic', 'other testing method', 'pressure test', 'x ray radiography', 'welding', 'liquid penetrant' or 'magnetic particle'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'other testing method'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'performed by customer' and a **description** of either 'false' or 'true'.

**WR4:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** in which the **kind** is exactly one **document\_kind** with **product\_data\_type** of 'inspection requirements'.

### 5.2.3.1.233 sprue

A **sprue** is a type of **casting\_feature\_definition** that is an opening from the top of the mould to the gating system and is where the molten metal enters the mould cavity.

NOTE A **Sprue** is defined according to 4.2.3.10.

EXPRESS specification:

\*)

```

ENTITY sprue
  SUBTYPE OF (gating_design_feature);
  WHERE

  (* --sprues to item_size (as sprue_dimensions) -- *)

  WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'sprue size usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) =1 )))) >=1;

  (* **sprues to filter (as metal_flow_filter) * *)

  wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'filter reference usage') AND
    ('CAST_PARTS_SCHEMA.FILTER' IN TYPEOF(sar.relying_shape_aspect))
    )) <=1 )))) >=1;

  (* ***** sprues to well (as sprue_well) ***** *)

  wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'well reference usage') AND
    ('CAST_PARTS_SCHEMA.WELL' IN TYPEOF(sar.relying_shape_aspect))
    )) >=0 )))) >=1;

  (* ----- sprue core ----- *)

  wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'sprue core')))) =1 ))))=1;

```

```

(* --sprues to item_size (as pouring_cup) -- *)

WR5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue pouring cup size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 ))) >=1;
END_ENTITY; -- sprue
(*

```

### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'sprue occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'sprue size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'sprue occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'filter reference usage' in which the **relating\_shape\_aspect** is a **filter**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'sprue occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'well reference usage' in which the **relating\_shape\_aspect** is a **well**.

**WR4:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'sprue core'.

**WR5:** Exactly one **shape\_aspect** with a **description** of 'sprue occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'sprue pouring cups size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.234 sprue\_and\_runner\_mould

A **sprue\_and\_runner\_mould** is a type of **investment\_design\_feature** that is the die that forms the sprue and the runner in wax or EPS to which the ingate and pattern(s) is attached.

NOTE A **Sprue\_and\_runner\_mould** is defined according to 4.2.311.

#### EXPRESS specification:

\*)

```

ENTITY sprue_and_runner_mould
  SUBTYPE OF (investment_design_feature);
  WHERE

  (* --sprue_and_runner_mould to sprue ( as pouring_sprue) -- *)

  WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue and runner mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'sprue reference usage') AND
    ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relateing_shape_aspect))
    )) =1 )))) >=1;

  (* -- sprue_and_runner_mould to runner (as gate_runner) -- *)

  wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue and runner mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'runner reference usage') AND
    ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(sar.relateing_shape_aspect))
    )) >=1 )))) >=1;

  (* -- sprue_and_runner_mould to ingate (as grigging_component) -- *)

  wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='sprue and runner mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'ingate reference usage') AND
    ('CAST_PARTS_SCHEMA.INGATE' IN TYPEOF(sar.relateing_shape_aspect))
    )) >=1 )))) >=1;

```

```

(* -- sprue_and_runner_mould to ingate (as mould_assembly) -- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue and runner mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment mould reference usage') AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_MOULD'
  IN
  TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

END_ENTITY; -- sprue_and_runner_mould
(*

```

### Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'sprue and runner mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'sprue reference usage' in which the **relating\_shape\_aspect** is a **sprue**.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'sprue and runner mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'runner reference usage' in which the **relating\_shape\_aspect** is a **runner**.

**WR3:** Exactly one **shape\_aspect** with a **description** of 'sprue and runner mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'ingate reference usage' in which the **relating\_shape\_aspect** is a **ingate**.

**WR4:** Exactly one **shape\_aspect** with a **description** of 'sprue and runner mould occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **shape\_defining\_relationship** in with a **description** of 'investment mould reference usage' in which the **relating\_shape\_aspect** is a **investment\_mould**.

### 5.2.3.1.235 stop\_pin

A **stop\_pin** is a type of **casting\_feature\_definition** that is a pin(s) attached to the Ejector\_plate that limits the retraction travel distance and may also limit the ejection travel distance.

NOTE A **Stop\_pin** is defined according to 4.2.313.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

```
ENTITY stop_pin
  SUBTYPE OF (ejector_design_feature);
  WHERE

(* ----- stop_pin to item_size (as size) ----- *)

WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='stop pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'stop pin size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1;
END_ENTITY; -- stop_pin
(*
```

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'stop pin occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'stop pin size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

### 5.2.3.1.236 surface\_finish

A **surface\_finish** is a type of **manufacturing\_activity** that that identifies the activity of applying a finish to the cast part.

NOTE A **Surface\_finish** is defined according to 4.2.316.

EXPRESS specification:

\*)

```
ENTITY surface_finish
  SUBTYPE OF (non_casting_activity);
END_ENTITY;
(*
```



### 5.2.3.1.237 tolerance\_inspection

A **tolerance\_inspection** is a type of **inspection\_or\_test\_result** that is the outcome of a measurement and comparing the resultant tolerance to the design tolerance requirement.

NOTE A **Tolerance\_inspection** is defined according to 4.2.324.

#### EXPRESS specification:

\*)

```

ENTITY tolerance_inspection
  SUBTYPE OF (inspection_or_test_result);
  WHERE

(* --tolerance_inspection to numeric_parameter (as tolerance_result)-- *)

wr1:(NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY( it <* pdr.used_representation.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
    (it.name = 'tolerance result') )) >=1)) ) =0));

(* --tolerance_inspection to tolerance_requirement (as requirements)- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
  IN TYPEOF(pdr.related_property_definition)
  )) >=1);
END_ENTITY; -- tolerance_inspection
(*)

```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'tolerance result'.

**WR2:** Exactly one reference that is the **related\_property\_definition** in exactly one **property\_definition\_relationship** in which the **relating\_property\_definition** is an **tolerance\_requirement**.

### 5.2.3.1.238 tolerance\_requirement

A **tolerance\_requirement** is a type of **customer\_casting\_requirement** that is as design requirement for tolerances on a part.

NOTE A **Tolerance\_requirement** is defined according to 4.2.328.

EXPRESS specification:

\*)

```
ENTITY tolerance_requirement
    SUBTYPE OF (customer_casting_requirement);
WHERE
```

(\* ----- allowance\_value ----- \*)

```
wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND
    (it.name = 'dimension method') AND
    (it.description IN ['commercial practice',
    'specification data','drawing data'] )
    ) ))=1 ))) = 0);
```

(\* ----- gaging\_fixtures\_available----- \*)

```
wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND
    (it.name = 'gaging fixtures available') AND
    (it.description IN ['TRUE','FALSE'] )
    ) ))=1 ))) = 0);
```

(\* ----- to\_dimensional\_tolerance(as tolerance\_based\_on)----- \*)

```
wr3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND
    (it.name = 'tolerance based one') AND
    (it.description IN ['mismatch','dimensional','geometric'] )
    ) ))=1 ))) = 0);
```

(\* --tolerance\_requirement to checking\_aid\_tool(as gaging\_requirement)-- \*)

```
wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.CHECKING_AID_TOOL'
    IN TYPEOF(pdr.related_property_definition)
    )))>=0);
```

```
(* --- tolerance_requirement to mismatch_tolerance( as mismatch)-- *)

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.MISMATCH_TOLERANCE'
IN TYPEOF(pdr.related_property_definition)
)))<=1);

(* - tolerance_requirement to specification (as dimension_specification)- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
(adr.assigned_document.kind.product_data_type=
'dimension specification') )) <=1;
END_ENTITY; -- tolerance_requirement
(*
```

#### Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'dimension method' and a **description** of either 'commercial practice', 'specification data' or 'drawing data'.

**WR2:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'gaging fixtures available' and a **description** of either 'true' or 'false'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'tolerance based on' and a **description** of either 'mismatch', 'dimensional' or 'geometric'.

**WR4:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** in which the **related\_property\_definition** is a **checking\_aid\_tool**.

**WR5:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** in which the **related\_property\_definition** is a **mismatch\_tolerance**.

**WR6:** Exactly one reference that is the **items** in exactly one **applied\_document\_reference** that is of type **document\_reference** in which the **assigned\_document** is an **document** that references through the **kind** attribute **document\_type** with **product\_data\_type** of 'dimension specification'.

### 5.2.3.1.239 tool\_verification

A **tool\_verification** is a type of **casting\_method** that is the process or procedure used to validate the foundry tool to ensure the customer and metalcaster specifications are met.

NOTE A **Tool\_verification** is defined according to 4.2.332.

EXPRESS specification:

\*)

```
ENTITY tool_verification
  SUBTYPE OF (casting_method);
  WHERE
```

(\* -----tool\_verification to core\_assembly (as core\_fit)----- \*)

```
wr1: (SIZEOF(QUERY(am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
  ('CAST_PARTS_SCHEMA.CORE_ASSEMBLY' IN TYPEOF(am.related_method))))
  <=1);
```

(\* -----tool\_verification to core\_print (as verify\_clearance)--\*)

```
wr2: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(am)) AND
  (SIZEOF(QUERY( cp <* am.items |
  ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(cp)))) >=1 ))))=0));
```

(\* --tool\_verification to design\_part (as based\_on\_part\_design)- \*)

```
wr3: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(am)) AND
  (SIZEOF(QUERY( cp <* am.items |
  ('CAST_PARTS_SCHEMA.DESIGN_PART' IN TYPEOF(cp)))) =1 ))))=0));
```

(\* -----tool\_verification to shape\_dimensions (as verify)----- \*)

```
wr4: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(am)) AND
  (SIZEOF(QUERY( cp <* am.items |
  ('CAST_PARTS_SCHEMA.SHAPE_DIMENSION' IN TYPEOF(cp)))) =1 ))))=0));
```

```
(* tool_verification to tolerance_requirement (as customer_requirement)--*)

wr5: (NOT (SIZEOF(QUERY(am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(am)) AND
(SIZEOF(QUERY(cp <* am.items |
('CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
IN TYPEOF(cp)))) >=1 ))))=0));

END_ENTITY; -- tool_verification
(*
```

### Formal propositions:

**WR1:** Exactly one reference that is the **related\_method** in exactly one **action\_method\_relationship** in which the **relating\_method** is an **core\_assembly**.

**WR2:** Exactly one reference that is the **method** in exactly one **action\_request\_solution** that has exactly one reference that is the **request** in exactly one **version\_action\_request** that has exactly one reference that is the **assigned\_action\_request** in exactly one **action\_request\_assignment** is of type **applied\_action\_request\_assignment** in which the **items** is exactly one **core\_print**.

**WR3:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** in which the **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **product\_definition** of type **design\_part**.

**WR4:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** in which the **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **property\_definition** of type **shape\_dimension**.

**WR5:** Exactly one reference that is the **assigned\_action** in exactly one **action\_assignment** in which the **action\_assignment** is of type **applied\_action\_assignment** in which the **items** is a **property\_definition** of type **tooling\_requirement**.

### 5.2.3.1.240 tooling\_process

The **tooling\_process** is a type of **casting\_activity** that is the requirement for foundry tooling to be produced by a pattern shop or tool and die shop.

NOTE A **Tooling\_process** is defined according to 4.2.334.

EXPRESS specification:

\*)

```
ENTITY tooling_process
  SUBTYPE OF (casting_activity);
  WHERE
```

(\* ----- number\_of\_impressions ----- \*)

```
wr1: (NOT (SIZEOF (QUERY (ap <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  (( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of impressions')
  ))) =1 ))) =0 ))) =0));
```

(\* ----- shrinkage factor ----- \*)

```
wr2: (NOT (SIZEOF (QUERY (ap <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  ((SIZEOF ([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) =2) AND
  (it.name = 'shrinkage factor')
  ))) =1 ))) =0 ))) =0));
```

(\* - process) - \*)

```
wr3: (NOT (SIZEOF (QUERY (ap <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name='process method') AND
  (it.description IN [
  'casting process',
  'non permanent moulding process',
  'permanent moulding process'] )
  ))) >=0 ))) =0 )) =0));
```

```

(* ---- tooling_proces to core (as cores_to_make) ----- *)

wr4:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.CORE' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* ---permanent_mould_process to gating_system (as required_gating) --- *)

wr5:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM' IN TYPEOF(eds.definition))
  ))) =1 ))))=0));

(* -- to customer_casting_requirement (as customer_requirements) - *)

wr6:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT' IN TYPEOF (eds))
  ))) >=1 ))))=0));

(* ---- permanent_mould_process to design_part (as part_geometry) --- *)

wr7:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.DESIGN_PART' IN TYPEOF (eds))
  ))) =1 ))))=0));

(* ----- to tool_verification (as quality_assurance) ----- *)

wr8:(SIZEOF(QUERY(amr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
  ('CAST_PARTS_SCHEMA.TOOL_VERIFICATION' IN TYPEOF(amr.related_method)
  )))=1);

END_ENTITY; -- tooling_process
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **measure\_representation\_item** and **count\_measure** with a **name** of 'number of impressions'.

**WR2:** The implicit representation of the feature shall have at most one **representation\_item** of type **measure\_representation\_item** and **length\_measure\_with\_unit** with a **name** of 'shrinkage factor'.

**WR3:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** and **count\_measure** with a **name** of 'process method' and **description** of 'casting process', 'non permanent moulding process', 'permanent moulding process'.

**WR4:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment** in which the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is a **property\_definition** in which the **definition** attribute references a **core**.

**WR5:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment** in which the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is a **property\_definition** in which the **definition** attribute references a **gating\_system**.

**WR6:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment** in which the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is a **property\_definition** that is of type **customer\_casting\_requirement**.

**WR7:** Exactly one reference that is the **assigned\_action\_method** in exactly one **action\_method\_assignment** in which the **action\_method\_assignment** is of type **applied\_action\_method\_assignment** in which the **items** is a **product\_definition** that is of type **design\_part**.

**WR8:** Exactly one reference that is the **relating\_property\_definition** in exactly one **property\_definition\_relationship** in which the **related\_property\_definition** is a **tool\_verification**.

### 5.2.3.1.241 value\_range

A **value\_range** is a type of **compound\_representation\_item** that specifies a range of values defined by two **measure\_representation\_items**.

NOTE This application interpreted model is harmonized with the same object from paragraph 5.2.3.2.88 of ISO 10303-240:2005.



EXPRESS specification:

```

*)

ENTITY value_range
  SUBTYPE OF (compound_representation_item);

WHERE

wr1:  SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
  ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
  ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
  ))=2;

wr2:  SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
  ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
  AND (mri.name='lower limit')) )=1;

wr3:  SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
  ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
  AND (mri.name='upper limit')) )=1;

wr4:  SIZEOF(QUERY( i1 <* SELF.item_element |
  ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i1)) AND
  (SIZEOF ( QUERY (i2 <* SELF.item_element |
  ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i2)) AND
  (i1 :<>: i2) AND
  (i1\measure_with_unit.unit_component :=:
  i2\measure_with_unit.unit_component)
  ) ) = 1 ))) = 2 ;

END_ENTITY;
(*

```

Formal propositions:

**WR1:** The **set\_representation\_item** shall have exactly two **representation\_items** of the same type in its set that are **measure\_representation\_items**.

**WR2:** One of the **representation\_items** in the set specified by **item\_element** shall have a name of 'upper limit',

**WR3:** One of the **representation\_items** in the set specified by **item\_element** shall have a name of 'upper limit',

**WR4:** If the set specified by **item\_element** consists of **measure\_representation\_items**, then the **measure\_representation\_items** shall point to the same instance **unit\_component**.

### 5.2.3.1.242 vent

The **vent** is a type of **gating\_design\_feature** that is the part of the gating system that most restricts or regulates the flow of metal into the mould cavity.

NOTE A **Vent** is defined according to 4.2.336.

EXPRESS specification:

```

*)
  ENTITY vent
    SUBTYPE OF (gating_design_feature);
    WHERE

(* -----   type of vent ----- *)

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'type of vent') AND
    ((it.description IN ['air','gas'] )) )) ) ) =1 )) )>=1;

(* -----   vent to item_size (as core_vent_size) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='vent occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'vent reference usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relater_shape_aspect))
  )) =1 )))) >=1;

(* -----   vent to location_element (as vent_placement)----- *)

wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='location')))=1;
  END_ENTITY; -- vent
(*

```

Formal propositions:

**WR1:** Exactly one **representation\_item** used for the implicit representation shall be of type **descriptive\_representation\_item** with a **name** of 'type of vent' and a **description** of either 'air' or 'gas'.

**WR2:** Exactly one **shape\_aspect** with a **description** of 'vent occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'vent reference usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR3:** There shall be exactly one reference through the **definition** attribute to a **product\_definition** that is referenced by **property\_definition\_representation** through the **definition** attribute and references a **location\_shape\_representation** through the **used\_representation** that has a **name** of 'location'.

### 5.2.3.1.243 view\_reference

A **view\_reference** is a type of **document\_usage\_constraint** that is identification, within a drawing, of a specific area of interest.

NOTE 1 This application interpreted model is harmonized with the same object from paragraph 5.2.3.1.89 of ISO 10303-240:2005.

NOTE 2 A **View\_reference** is defined according to 4.2.337.

#### EXPRESS specification:

```

*)
ENTITY view_reference
    SUBTYPE OF (document_usage_constraint);
WHERE

    wr1: (NOT(SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF.source,
        'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
        ('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN TYPEOF(dr)) |
        (SIZEOF(QUERY ( d <* adr.items |
        ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) )
        >= 0) )) = 0));

    wr2: 'CAST_PARTS_SCHEMA.DESIGN_REFERENCE' IN TYPEOF (SELF.source);

    wr3: SELF.subject_element IN ['sheet','view','zone'];

END_ENTITY; -- view_reference
(*

```

Formal propositions:

**WR1:** There shall be one or more **assigned\_document** in exactly one **applied\_document\_reference** that has in the list of **items** one or more **process\_planning\_activity**.

**WR2:** There shall be exactly one **document\_usage\_constraint\_assignment** referenced through **assigned\_document\_usage** that has one item in the list of **items** that references a **process\_plan\_activity**.

**WR3:** The **subject\_element** is of 'sheet', 'view', or 'zone'.

### 5.2.3.1.244 well

A **well** is a type of **casting\_feature\_definition** that is a basin like reservoir that is located on the drag side of the mould beneath the sprue and riser and at the same center point.

NOTE A **Well** is defined according to 4.2.338.

EXPRESS specification:

\*)

ENTITY well

SUBTYPE OF (gating\_design\_feature);  
WHERE

(\* ----- well to item\_size (as well\_dimensions) ----- \*)

```
WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='well occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'well size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) =1 ))) >=1;
```

(\* ----- well to parting\_surface (as drag\_surface) ----- \*)

```
wr2: SIZEOF(QUERY( pdr <* get_property_definition_representations(SELf) |
  ('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF(pdr.used_representation)) )) = 1;
```

END\_ENTITY; -- well

(\*

Formal propositions:

**WR1:** Exactly one **shape\_aspect** with a **description** of 'well occurrence' is referenced by a **product\_definition** through the **definition** attribute that is of type **product\_definition\_shape** that is referenced by a **shape\_aspect** through the **of\_shape** attribute that is the **related\_shape\_aspect** in exactly one **feature\_component\_relationship** in with a **description** of 'well size usage' in which the **relating\_shape\_aspect** is a **casting\_feature\_size**.

**WR2:** Exactly one **representation** used for the implicit representation shall be of type **parting\_surface**.

## 5.2.3.2 Cast parts schema imported entity modifications

### 5.2.3.2.1 action\_relationship

The base definition of the **action\_relationship** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **action\_relationship** entity:

— **project\_order\_tracking\_relationships** (See 5.2.4.25).

### 5.2.3.2.2 application\_context

The base definition of the **application\_context** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application\_context** entity:

— **application\_context\_requires\_ap\_definition** (See 5.2.4.1).

### 5.2.3.2.3 application\_protocol\_definition

The base definition of the **application\_protocol\_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application\_protocol\_definition** entity:

— **application\_context\_requires\_ap\_definition** (See 5.2.4.1).

#### 5.2.3.2.4 approval\_status

The base definition of the **approval\_status** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **approval\_status** entity:

- restrict\_approval\_status (See 5.2.4.26).

#### 5.2.3.2.5 chamfer

The base definition of the **chamfer** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **chamfer** entity:

- chamfer\_requires\_faces (See 5.2.4.3).

#### 5.2.3.2.6 chamfer\_offset

The base definition of the **chamfer\_offset** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **chamfer\_offset** entity:

- chamfer\_offset\_requires\_faces (See 5.2.4.4).

#### 5.2.3.2.7 characterized\_object

The base definition of the **characterized\_object** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **characterized\_object** entity:

- subtype\_mandatory\_characterized\_object (See 5.2.4.30).

### 5.2.3.2.8 directed\_action

The base definition of the **directed\_action** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

#### Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **directed\_action** entity:

- project\_order\_requires\_approval (See 5.2.4.24).
- project\_order\_tracking\_relationships (See 5.2.4.25).

### 5.2.3.2.9 edge\_round

The base definition of the **edge\_round** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

#### Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **edge\_round** entity:

- edge\_round\_requires\_faces (See 5.2.4.16).

### 5.2.3.2.10 externally\_defined\_feature\_definition

The base definition of the **externally\_defined\_feature\_definition** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

#### Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **externally\_defined\_feature\_definition** entity:

- restrict\_externally\_defined\_feature\_definition (See 5.2.4.27).

### 5.2.3.2.11 instanced\_feature

The base definition of the **instanced\_feature** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

#### Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **instanced\_feature** entity:

- casting\_feature\_life\_cycle (See 5.2.4.2).

### 5.2.3.2.12 product

The base definition of the **product** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product** entity:

- product\_requires\_version (See 5.2.4.22).

### 5.2.3.2.13 product\_definition

The base definition of the **product\_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product\_definition** entity:

- material\_is\_specified\_for\_part (See 5.2.4.19).

### 5.2.3.2.14 product\_definition\_formation

The base definition of the **product\_definition\_formation** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product\_definition\_formation** entity:

- part\_requires\_project\_order (See 5.2.4.20);
- part\_to\_approval (See 5.2.4.21);
- product\_requires\_version (See 5.2.4.22);
- product\_version\_requires\_security\_classification (See 5.2.4.23).

### 5.2.3.2.15 product\_definition\_representation

The base definition of the **product\_definition\_representation** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.



Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product\_definition\_representation** entity:

- chamfer\_requires\_faces (See 5.2.4.3);
- chamfer\_offset\_requires\_faces (See 5.2.4.4);
- edge\_round\_requires\_faces (See 5.2.4.16).

**5.2.3.2.16 security\_classification\_level**

The base definition of the **security\_classification\_level** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 applies to the **security\_classification\_level** entity:

- restrict\_security\_classification\_level (see 5.2.4.28).

**5.2.3.2.17 transition\_feature**

The base definition of the **transition\_feature** entity is given in ISO 10303-522. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **transition\_feature** entity:

- transition\_feature\_life\_cycle(See 5.2.4.31).
- transition\_feature\_on\_part\_boundary (See 5.2.4.32).

**5.2.4 Cast parts schema rules****5.2.4.1 application\_context\_requires\_ap\_definition**

The **application\_context\_requires\_ap\_definition** rule specifies that each instance of **application\_context** shall be referenced by exactly one **application\_protocol\_definition** that specifies this part of ISO 10303.

EXPRESS specification:

```
*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE
  WR1: SIZEOF (QUERY (ac <* application_context |
    NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
      (ac :=: apd.application)
      AND
      (apd.application_interpreted_model_schema_name =
        'cast parts schema')))) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

**application\_context:** the set of all instances of **application\_context** entities.

**application\_protocol\_definition:** the set of all instances of **application\_protocol\_definition** entities.

Formal propositions:

**WR1:** For each instance of **application\_context**, there shall be exactly one instance of **application\_protocol\_definition** that references the instance of **application\_context** as its **application** with a value of 'cast parts schema' as its **application\_interpreted\_model\_schema\_name**.

### 5.2.4.2 casting\_feature\_life\_cycle

The **casting\_feature\_life\_cycle** rule specifies that each instance of **casting\_instanced\_feature** shall be defined for the manufacturing planning stage of the part on which it is specified.

EXPRESS specification:

```
*)
RULE casting_feature_life_cycle FOR
  (casting_instanced_feature);
WHERE
  WR1: SIZEOF (QUERY (mf <* casting_instanced_feature |
    NOT (mf.of_shape.definition.frame_of_reference.life_cycle_stage =
      'manufacturing planning')))) = 0;
END_RULE;
(*
```

Argument definitions:

**casting\_instanced\_feature:** the set of all instances of **casting\_instanced\_feature** entities.

Formal propositions:

**WR1:** For each instance of **casting\_instanced\_feature**, the **life\_cycle\_stage** of the **product\_definition** for which it is defined has a value of 'manufacturing planning'.

### 5.2.4.3 chamfer\_requires\_faces

The **chamfer\_requires\_faces** rule specifies that all instances of **chamfer** require **face\_shape\_representation**.

EXPRESS specification:

```
*)
RULE chamfer_requires_faces FOR (chamfer,
    property_definition_representation);
WHERE
    wr1: SIZEOF(QUERY ( er <* chamfer |
        (SIZEOF (QUERY (pd <* USEDIN (er,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
            NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
                'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF (pdr.used_representation)) AND
                (pdr.used_representation.name = 'chamfer face'))=1)))=1)))=0;

END_RULE;
(*
```

Argument definitions:

**chamfer:** the set of all instances of **chamfer** entities.

**property\_definition\_representation:** the set of all instances of **property\_definition\_representation** entities.

Formal propositions:

**WR1:** The **chamfer** shall have exactly one **face\_shape\_representations** in the role of the chamfer face.

### 5.2.4.4 chamfer\_offset\_requires\_faces

The **chamfer\_offset\_requires\_faces** rule specifies that all instances of **chamfer\_offset** require **face\_shape\_representation**.

EXPRESS specification:

```
*)
RULE chamfer_offset_requires_faces FOR (chamfer_offset,
```

```

        property_definition_representation);
WHERE
  WR1: SIZEOF(QUERY ( co <* chamfer_offset |
    (( (co.description = 'first offset')) AND
    (SIZEOF(QUERY ( pd <* USEDIN(co,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND
    (pdr.used_representation.name = 'first chamfer face')) )) = 1)) ))
    = 1)) ))=0;

  WR2: SIZEOF(QUERY ( co <* chamfer_offset |
    (( (co.description = 'second offset')) AND
    (SIZEOF(QUERY ( pd <* USEDIN(co,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
    IN TYPEOF(pdr.used_representation)) AND
    (pdr.used_representation.name = 'second chamfer face')) )) = 1)) ))
    = 1)) ))=0;

END_RULE;
(*

```

Argument definitions:

**chamfer\_offset:** the set of all instances of **chamfer\_offset** entities.

**property\_definition\_representation:** the set of all instances of **property\_definition\_representation** entities.

Formal propositions:

**WR1:** If the **description** of the **chamfer\_offset** is 'first offset', the **chamfer\_offset** shall have at most one one **face\_shape\_representation** with a **name** of 'first face shape' or if the **description** of the **chamfer\_offset** is 'second offset', the **chamfer\_offset** shall have at most one one **face\_shape\_representation** with a **name** of 'second face shape'.

### 5.2.4.5 composition\_element\_requires\_name

The **composition\_element\_requires\_name** rule specifies that all instances of **product\_definition** that has a **product\_definition\_context** with a **name** of 'material composition definition' shall have an element name equal to names in the chemical periodic table.

EXPRESS specification:

\*)

```

RULE composition_element_requires_name FOR (
    product_definition,product_definition_formation);

WHERE

wr1: SIZEOF(QUERY ( pd <* product_definition |
    ((pd.frame_of_reference.name='material composition definition') AND
    (NOT((SIZEOF(QUERY( pdf <* product_definition_formation |
    ((pd.formation ::= pdf) AND
    (pdf.of_product.name IN [
'Actinium', 'Aluminum', 'Americium', 'Antimony', 'Argon', 'Arsenic', 'Astatine',
'Barium', 'Berkelium', 'Beryllium', 'Bismuth', 'Bohrium', 'Boron', 'Bromine',
' Cadmium', 'Caesium', 'Calcium', 'Californium', 'Carbon', 'Cerium', 'Chlorine',
'Chromium', 'Cobalt', 'Copper', 'Curium', 'Darmstadtium', 'Dubnium', 'Dysprosium',
'Einsteinium', 'Erbium', 'Europium', 'Fermium', 'Fluorine', 'Francium',
'Gadolinium', 'Gallium', 'Germanium', 'Gold', 'Hafnium', 'Hassium', 'Helium',
'Holmium', 'Hydrogen', 'Indium', 'Iodine', 'Iridium', 'Iron', 'Krypton',
'Lanthanum', 'Lawrencium', 'Lead', 'Lithium', 'Lutetium', 'Magnesium',
'Manganese', 'Meitnerium', 'Mendelevium', 'Mercury', 'Molybdenum', 'Neodymium',
'Neon', 'Neptunium', 'Nickel', 'Niobium', 'Nitrogen', 'Nobelium', 'Osmium',
'Oxygen', 'Palladium', 'Phosphorus', 'Platinum', 'Plutonium', 'Polonium',
'Potassium', 'Praseodymium', 'Promethium', 'Protactinium', 'Radium', 'Radon',
'Rhenium', 'Rhodium', 'Roentgenium', 'Rubidium', 'Ruthenium', 'Rutherfordium',
'Samarium', 'Scandium', 'Seaborgium', 'Selenium', 'Silicon', 'Silver', 'Sodium',
'Strontium', 'Sulfur', 'Tantalum', 'Technetium', 'Tellurium', 'Terbium',
'Thallium', 'Thorium', 'Thulium', 'Tin', 'Titanium', 'Tungsten', 'Ununbium',
'Ununhexium', 'Ununoctium', 'Ununpentium', 'Ununquadium', 'Ununtrium',
'Uranium', 'Vanadium', 'Xenon', 'Ytterbium', 'Yttrium', 'Zinc', 'Zirconium']
)) )) =1 )) )) )) =0;

END_RULE; --composition_element_requires_name
(*)

```

Argument definitions:

**product\_definition** the set of all instances of **product\_definition** entities.

**product\_definition\_formation**: the set of all instances of **product\_definition\_formation** entities.

Formal propositions:

**WR1:** If the **product\_definition** references a **product\_definition\_context** with **name** of 'material composition definition' then the **product\_definition** shall reference a **product\_definition\_formation** that references a **product** with a **name** of 'Actinium', 'Aluminum', 'Americium', 'Antimony', 'Argon', 'Arsenic', 'Astatine', 'Barium', 'Berkelium', 'Beryllium', 'Bismuth', 'Bohrium', 'Boron', 'Bromine', 'Cadmium', 'Caesium', 'Calcium', 'Californium', 'Carbon', 'Cerium', 'Chlorine', 'Chromium', 'Cobalt', 'Copper', 'Curium', 'Darmstadtium', 'Dubnium', 'Dysprosium', 'Einsteinium', 'Erbium', 'Europium', 'Fermium',

'Fluorine', 'Francium', 'Gadolinium', 'Gallium', 'Germanium', 'Gold', 'Hafnium', 'Hassium', 'Helium', 'Holmium', 'Hydrogen', 'Indium', 'Iodine', 'Iridium', 'Iron', 'Krypton', 'Lanthanum', 'Lawrencium', 'Lead', 'Lithium', 'Lutetium', 'Magnesium', 'Manganese', 'Meitnerium', 'Mendelevium', 'Mercury', 'Molybdenum', 'Neodymium', 'Neon', 'Neptunium', 'Nickel', 'Niobium', 'Nitrogen', 'Nobelium', 'Osmium', 'Oxygen', 'Palladium', 'Phosphorus', 'Platinum', 'Plutonium', 'Polonium', 'Potassium', 'Praseodymium', 'Promethium', 'Protactinium', 'Radium', 'Radon', 'Rhenium', 'Rhodium', 'Roentgenium', 'Rubidium', 'Ruthenium', 'Rutherfordium', 'Samarium', 'Scandium', 'Seaborgium', 'Selenium', 'Silicon', 'Silver', 'Sodium', 'Strontium', 'Sulfur', 'Tantalum', 'Technetium', 'Tellurium', 'Terbium', 'Thallium', 'Thorium', 'Thulium', 'Tin', 'Titanium', 'Tungsten', 'Ununbium', 'Ununhexium', 'Ununoctium', 'Ununpentium', 'Ununquadium', 'Ununtrium', 'Uranium', 'Vanadium', 'Xenon', 'Ytterbium', 'Yttrium', 'Zinc' or 'Zirconium'.

### 5.2.4.6 customer\_order\_rules

The **customer\_order\_rules** rule specifies that all instances of **customer\_order** shall have these constraints on instantiations

#### EXPRESS specification:

\*)

```
RULE customer_order_rules FOR (directed_action);
WHERE
```

```
(* ----- customer_order to date (as delivery_date) ----- *)
```

```
wr1: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'customer order') AND
    (NOT (SIZEOF (QUERY (ada <* USEDIN (da,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
    (ada.role.name='delivery date'))=1))))=0;
```

```
(* ----- production_status ----- *)
```

```
wr2: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'customer order') AND
    (NOT (SIZEOF (QUERY (ap <* USEDIN (da,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
    (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (NOT (SIZEOF (QUERY (rep <* apr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF (rep)) AND
    (rep.name='production status') AND
    (rep.description IN ['current', 'experimental', 'prototype', 'new'] )
    )))=1))))=0))))=0))))=0;
```

```

(* ---customer_order to first_article (as first_article_requirements) -- *)

wr3: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT(SIZEOF(QUERY( fr <* da.directive.requests |
    'CAST_PARTS_SCHEMA.FIRST_ARTICLE' IN TYPEOF (fr))>=1))))=0;

(* - customer_order to inspection_plan (as inspection_procedure) ----- *)

wr4: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT(SIZEOF(QUERY(ada <* USEDIN(da,
    'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATED_ACTION') |
    ('CAST_PARTS_SCHEMA.INSPECTION_PLAN' IN TYPEOF(ada.relatng_action)
    ))=1))))=0;

(* ----- customer_order to request_for_quotation (as RFQ) ----- *)

wr5: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT(SIZEOF(QUERY( fr <* da.directive.requests |
    'CAST_PARTS_SCHEMA.REQUEST_FOR_QUOTATION' IN TYPEOF (fr))<=1))))=0;

(* ----- customer_order to ordered_part (as quantity_ordered) ----- *)

wr6: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT(SIZEOF(QUERY(ada <* USEDIN(da,
    'CAST_PARTS_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
    ('CAST_PARTS_SCHEMA.ORDERED_PART' IN TYPEOF(ada)
    ))=1))))=0;

(* ---- customer_order to casting_process (as casting_method) ----- *)

wr7: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT(SIZEOF(QUERY(ap <* USEDIN(da,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (NOT(SIZEOF(QUERY(apr <* USEDIN(ap,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (NOT(SIZEOF(QUERY(rep <* apr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(rep)) AND
    (rep.name='casting method') AND
    (rep.description IN [
    'core presentation',
    'core setting',
    'die assembly',
    'die handling',
    'die temperature control',
    'furnace loading',
    'gate system removal',
    'grinding',
    'inspection',

```

ISO 10303-223:2008 (E)

```
'ladle loading',
'loading injection system',
'melt raw material',
'mould closing',
'mould making process',
'package and shipping',
'part routing',
'pattern handling',
'tool handling',
'pouring process',
'refractor removal',
'removal from die',
'shakeout',
'slurry process',
'sand removal',
'wax removal',
'shell',
'green sand',
'no bake',
'lost form',
'vacuum',
'static',
'tilt',
'low pressure'] )
)))=1)))=0)))=0)))=0;

(* -----customer_order to person_in_organization (as customer) ----- *)
```

```
wr8: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT (SIZEOF (QUERY (ada <* USEDIN (da,
    'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    (ada.role.name='customer')))=1))))=0;
```

```
(* ----- customer_order to project_order (as initiated_order) ----- *)
```

```
wr9: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT (SIZEOF (QUERY (ada <* USEDIN (da,
    'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATED_ACTION') |
    (ada.relating_action.name='project order'
    ))=1))))=0;
```



```
(* ----customer_order to simulation_report (as simulation_run_result)--- *)

wr10: SIZEOF (QUERY (da <* directed_action |
  (da.name = 'customer order') AND
  (NOT (SIZEOF (QUERY (ada <* USEDIN (da,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (ada.assigned_document\document.kind.product_data_type=
      'simulation report'))=1))))=0;

END_RULE;
(*
```

### Argument definitions:

**directed\_action:** the set of all instances of **directed\_action** entities.

### Formal propositions:

**WR1:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by at least one **applied\_date\_assignment** through the **items** attribute, and the **applied\_date\_assignment** shall reference **date\_role** with **name** of 'delivery date' through the **role** attribute.

**WR2:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by at least one **action\_property** through the **definition** attribute, and the **action\_property** shall be referenced through the **property** attribute by an **action\_property\_representation** that references through the **representation** attribute a **representation** that references a **descriptive\_representation\_item** through the **items** attribute that has a **name** of 'production status' and a **description** of either 'current', 'experimental', 'prototype' or 'new'.

**WR3:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall reference through the **directive** attribute an **action\_directive** that shall reference through the **requests** attribute a **version\_action\_request** of type **first\_article**.

**WR4:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by through the **related\_action** attribute of an **action\_relationship** that references through the **relating\_action** an **action** of type **inspection\_plan**.

**WR5:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall reference through the **directive** attribute an **action\_directive** that shall reference through the **requests** attribute a **version\_action\_request** of type **request\_for\_quotation**.

**WR6:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by through the **assigned\_action** attribute of an **action\_assignment** that is of type **ordered\_part**.

**WR7:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by at least one **action\_property** through the **definition** attribute, and the **action\_property** shall be referenced through the **property** attribute by an **action\_property\_representation** that references through the **representation** attribute a **representation** that references a **descriptive\_representation\_item** through the **items** attribute that has a **name** of 'casting method' and a **description** of either 'core presentation', 'core setting', 'die assembly', 'die handling', 'die temperature control', 'furnace loading', 'gate system removal', 'grinding', 'inspection', 'ladle loading', 'loading injection system', 'melt raw material', 'mould closing', 'mould making process', 'package and shipping', 'part handling', 'pattern handling', 'pouring process', 'refractor removal', 'removal from die', 'shakeout', 'slurry process', 'sand removal', 'wax removal', 'shell', 'green sand', 'no bake', 'lost form', 'vacuum', 'static', 'tilt', 'low pressure'.

**WR8:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by at least one **applied\_person\_and\_organization\_assignment** through the **items** attribute, and the **applied\_person\_and\_organization\_assignment** shall reference **person\_and\_organization\_role** with **name** of 'customer' through the **role** attribute.

**WR9:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by through the **related\_action** attribute of an **action\_relationship** that references through the **relating\_action** an **action** with a name of 'project order'.

**WR10:** For each instance of **directed\_action** with a **name** of 'customer order' then the **directed\_action** shall be referenced by at least one **applied\_document\_reference** through the **items** attribute, and the **applied\_document\_reference** that is of type **document\_reference** that references through the **assigned\_document** a **document** that references through the **kind** attribute a **document\_type** with the **product\_data\_type** of 'simulation report'.

#### 5.2.4.7 **dependent\_instantiable\_approval\_status**

The **dependent\_instantiable\_approval\_status** rule specifies that all instances of **approval\_status** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
    NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**approval\_status:** the set of all instances of **approval\_status**.

Formal proposition:

**WR1:** For each instance of **approval\_status**, there shall be a reference to the **approval\_status** instance from an attribute of another entity.

### 5.2.4.8 dependent\_instantiable\_date

The **dependent\_instantiable\_date** rule specifies that all instances of **date** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date FOR (date);
WHERE
  WR1: SIZEOF (QUERY (dt <* date |NOT(SIZEOF (USEDIN (dt, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**date:** the set of all instances of **date**.

Formal proposition:

**WR1:** For each instance of **date**, there shall be a reference to the **date** instance from an attribute of another entity.

### 5.2.4.9 dependent\_instantiable\_named\_unit

The **dependent\_instantiable\_named\_unit** rule specifies that all instances of **named\_unit** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_named_unit FOR (named_unit);
WHERE
  WR1: SIZEOF (QUERY (nu <* named_unit |
    NOT (SIZEOF (USEDIN (nu, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**named\_unit:** the set of all instances of **named\_unit**.

ISO 10303-223:2008 (E)

Formal proposition:

**WR1:** For each instance of **named\_unit**, there shall be a reference to the **named\_unit** instance from an attribute of another entity.

#### 5.2.4.10 dependent\_instantiable\_precision\_qualifier

The **dependent\_instantiable\_precision\_qualifier** rule specifies that all instances of **precision\_qualifier** are dependent on the usage to define another entity.

EXPRESS specification:

\*)

```
RULE dependent_instantiable_precision_qualifier FOR (precision_qualifier);
WHERE
  WR1: SIZEOF (QUERY (pq <* precision_qualifier |
    NOT (SIZEOF (USEDIN (pq, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**precision\_qualifier:** the set of all instances of **precision\_qualifier**.

Formal proposition:

**WR1:** For each instance of **precision\_qualifier**, there shall be a reference to the **precision\_qualifier** instance from an attribute of another entity.

#### 5.2.4.11 dependent\_instantiable\_type\_qualifier

The **dependent\_instantiable\_type\_qualifier** rule specifies that all instances of **type\_qualifier** are dependent on the usage to define another entity.

EXPRESS specification:

\*)

```
RULE dependent_instantiable_type_qualifier FOR (type_qualifier);
WHERE
  WR1: SIZEOF (QUERY (tq <* type_qualifier |
    NOT (SIZEOF (USEDIN (tq, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**type\_qualifier:** the set of all instances of **type\_qualifier**.

Formal proposition:

**WR1:** For each instance of **type\_qualifier**, there shall be a reference to the **type\_qualifier** instance from an attribute of another entity.

#### 5.2.4.12 dependent\_instantiable\_uncertainty\_qualifier

The **dependent\_instantiable\_uncertainty\_qualifier** rule specifies that all instances of **uncertainty\_qualifier** are dependent on the usage to define another entity.

EXPRESS specification:

\*)

```

RULE dependent_instantiable_uncertainty_qualifier FOR
  (uncertainty_qualifier);
WHERE
  WR1: SIZEOF (QUERY (uq <* uncertainty_qualifier |
    NOT (SIZEOF (USEDIN (uq, '')) >= 1))) = 0;
END_RULE;
(*

```

Argument definition:

**uncertainty\_qualifier:** the set of all instances of **uncertainty\_qualifier**.

Formal proposition:

**WR1:** For each instance of **uncertainty\_qualifier**, there shall be a reference to the **uncertainty\_qualifier** instance from an attribute of another entity.

#### 5.2.4.13 dependent\_instantiable\_security\_classification\_level

The **dependent\_instantiable\_security\_classification\_level** rule specifies that all instances of **security\_classification\_level** are dependent on the usage to define another entity.

ISO 10303-223:2008 (E)

EXPRESS specification:

\*)

```
RULE dependent_instantiable_security_classification_level FOR
  (security_classification_level);
WHERE
  WR1: SIZEOF (QUERY (scl <* security_classification_level |
    NOT (SIZEOF (USEDIN (scl, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**security\_classification\_level:** the set of all instances of **security\_classification\_level**.

Formal proposition:

**WR1:** For each instance of **security\_classification\_level**, there shall be a reference to the **security\_classification\_level** instance from an attribute of another entity.

#### 5.2.4.14 dependent\_instantiable\_shape\_representation

The **dependent\_instantiable\_shape\_representation** rule specifies that all instances of **shape\_representation** are dependent on the usage to define another entity.

EXPRESS specification:

\*)

```
RULE dependent_instantiable_shape_representation FOR (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (sr <* shape_representation |
    NOT (SIZEOF (USEDIN (sr, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

**shape\_representation:** the set of all instances of **shape\_representation**.

Formal proposition:

**WR1:** For each instance of **shape\_representation**, there shall be a reference to the **shape\_representation** instance from an attribute of another entity.

### 5.2.4.15 document\_file\_rules

The **document\_file\_rules** rule specifies that all instances of **document\_file** shall have these constraints on instantiations

#### EXPRESS specification:

\*)

```
RULE document_file_rules FOR (document_file);
```

```
WHERE
```

```
(* --quality_acceptance_report to reporting_requirement (as applied_to) - *)
```

```
wr1: ((SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'quality acceptance report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
    (('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN TYPEOF(doc)) AND
    (SIZEOF(QUERY(rep <* doc.items |
      ('CAST_PARTS_SCHEMA.REPORTING_REQUIREMENT' IN TYPEOF(rep))
    ))>=1))))=0))))=0));
```

```
(* ----- Product_quality_report action_required ----- *)
```

```
wr2: (NOT(SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT(SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        (it.name = 'action required')) =1 ))) =0 )))) =0 )))) =0));
```

```
(* ----- Product_quality_report number_failed ----- *)
```

```
wr3: (NOT (SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT(SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'number failed')
        ) ) =1 ))) =0 )))) =0 )))) =0));
```

## ISO 10303-223:2008 (E)

```
(* Product_quality_report to inspection_or_test_result (as what_failed) -*)

wr4: ((SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type ='product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    ('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_RESULT' IN TYPEOF(pd))))=1)
  )))=0));

(* -- customer_design_deficiency_report list_of_results ----- *)

wr5: (NOT(SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type ='product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT(SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        (it.name = 'list of results')))) =1 ))) =0 ))) =0 )))) =0));

(* --- customer_tool_design_report to tooling (as designed_tool ----- *)

wr6: ((SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type ='customer tool design report') AND
  (SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT') |
    (NOT ('CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION'
      IN TYPEOF(pd.related_product)
    ))))=1 ))))=0));

END_RULE; -- document_file__rules
(*
```

### Argument definitions:

**document\_file:** the set of all instances of **document\_file** entities.



Formal propositions:

**WR1:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'quality acceptance report', there may be one or more references through the **assigned\_document** attribute of a **document\_reference** of type **applied\_document\_reference** that references through the **items** attribute a **property\_definition** of type **reporting\_requirement**.

**WR2:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'quality acceptance report', there may be one or more references through the **assigned\_document** attribute of a **document\_reference** of type **applied\_document\_reference** that references through the **items** attribute a **property\_definition** that is reference through the **definition** attribute of a **property\_definition\_representation** that references through the **used\_representation** attribute a **representation** that references through the **items** attribute a **descriptive\_representation\_item** with **name** of 'action required'.

**WR3:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'quality acceptance report', there may be one or more references through the **assigned\_document** attribute of a **document\_reference** of type **applied\_document\_reference** that references through the **items** attribute a **property\_definition** that is reference through the **definition** attribute of a **property\_definition\_representation** that references through the **used\_representation** attribute a **representation** that references through the **items** attribute a **measure\_representation\_item** with **name** of 'number failed' and of type **measure\_with\_unit** that references through the **value\_component** attribute a **count\_measure**.

**WR4:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'quality acceptance report', there may be one or more references through the **assigned\_document** attribute of a **document\_reference** of type **applied\_document\_reference** that references through the **items** attribute a **property\_definition** that is of type **inspection\_or\_test\_result**.

**WR5:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'customer design deficiency report', there may be one or more references through the **assigned\_document** attribute of a **document\_reference** of type **applied\_document\_reference** that references through the **items** attribute a **property\_definition** that is reference through the **definition** attribute of a **property\_definition\_representation** that references through the **used\_representation** attribute a **representation** that references through the **items** attribute a **descriptive\_representation\_item** with **name** of 'list of results'.

**WR6:** For each instance of **document\_file** that references through the **kind** attribute a **document\_type** that has a **product\_data\_type** of 'customer tool design report', there may be one or more references through the **relating\_document** attribute of a **document\_product\_association** that references through the **related\_product** a **product\_definition** of type **casting\_product\_definition**.

#### 5.2.4.16 **edge\_round\_requires\_faces**

The **edge\_round\_requires\_faces** rule specifies that all instances of **edge\_round** require **face\_shape\_representation**.

EXPRESS specification:

```

*)
RULE edge_round_requires_faces FOR
    (edge_round, property_definition_representation);
WHERE
    wr1: SIZEOF(QUERY ( er <* edge_round |
        (SIZEOF (QUERY (pd <* USEDIN (er,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
            NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
                'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF (pdr.used_representation)) AND
                (pdr.used_representation.name = 'edge round face')) =1 ))) =1 )))=0;

    wr2: SIZEOF(QUERY ( er <* edge_round |
        (SIZEOF (QUERY (pd <* USEDIN (er,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
            NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
                'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF (pdr.used_representation)) AND
                (pdr.used_representation.name = 'first face shape')) =1 ))) =1 )))=0;

    wr3: SIZEOF(QUERY ( er <* edge_round |
        (SIZEOF (QUERY (pd <* USEDIN (er,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
            NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
                'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF (pdr.used_representation)) AND
                (pdr.used_representation.name = 'second face shape')) =1 ))) =1 )))=0;
END_RULE;
(*

```

Argument definitions:

**edge\_round:** the set of all instances of **edge\_round** entities.

**property\_definition\_representation:** the set of all instances of **property\_definition\_representation** entities.

Formal propositions:

**WR1:** The **edge\_round** shall have exactly one **face\_shape\_representation** with a **name** of 'edge round face'.

**WR2:** The **edge\_round** shall have exactly one **face\_shape\_representation** with a **name** of 'first face shape'.

**WR3:** The **edge\_round** shall have exactly one **face\_shape\_representation** with a **name** of 'second face shape'.

### 5.2.4.17 externally\_defined\_class\_with\_known\_source\_requirement

The **externally\_defined\_class\_with\_known\_source\_requirement** rule specifies that each instance of **externally\_defined\_class** which references as the source a **known\_source** is contained in the set of items of exactly one **applied\_organization\_assignment** which references as its role an **organization\_role** with a name of 'library supplier'. This rule enforces the requirement for every library\_reference to have a supplier\_BSU.

EXPRESS specification:

```
*)
RULE externally_defined_class_with_known_source_requirement FOR
(externally_defined_class);
WHERE
  wr1: SIZEOF(QUERY ( edc <* externally_defined_class |
    (('CAST_PARTS_SCHEMA.KNOWN_SOURCE'
      IN TYPEOF(edc.source)) AND
      (NOT (SIZEOF(QUERY ( aoa <* USEDIN(edc.source,
        'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
          (aoa.role.name = 'library supplier')) = 1)))) <= 1;
END_RULE;
(*
```

Argument definitions:

**externally\_defined\_class:** the set of all instances of **externally\_defined\_class**

Formal propositions:

**WR1:** Each instance of **externally\_defined\_class** which references as the source a **known\_source** shall be member in the set of items of an **applied\_organization\_assignment**. This **organization\_assignment** shall reference as its role an **organization\_role** with a name of 'library supplier'.

### 5.2.4.18 machining\_feature\_life\_cycle

The **machining\_feature\_life\_cycle** rule specifies that each instance of **instanced\_feature** shall be defined for the manufacturing planning stage of the part on which it is specified.

EXPRESS specification:

```
*)
RULE machining_feature_life_cycle FOR
(instanced_feature);
```

© ISO 2008 — All rights reserved

1347

ISO 10303-223:2008 (E)

```
WHERE
  WR1: SIZEOF (QUERY (mf <* instanced_feature |
    NOT (mf.of_shape.definition.frame_of_reference.life_cycle_stage =
      'manufacturing planning')) = 0;
END_RULE;
(*
```

Argument definitions:

**instanced\_feature:** the set of all instances of **instanced\_feature** entities.

Formal propositions:

**WR1:** For each instance of **instanced\_feature**, the **life\_cycle\_stage** of the **product\_definition** for which it is defined has a value of 'manufacturing planning'.

### 5.2.4.19 material\_is\_specified\_for\_part

The **material\_is\_specified\_for\_part** rule specifies that every **product\_definition** that is not designated a material shall be related to a material designated **product\_definition** through the **make\_from\_usage\_option**.

EXPRESS specification:

```
*)
RULE material_is_specified_for_part FOR (product_definition,
  make_from_usage_option);
WHERE
wr1: SIZEOF(QUERY ( nmpd <* QUERY ( pd <* product_definition |
  (SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS'))
  = 0) ) |
  (NOT (SIZEOF(QUERY ( mfuo <* make_from_usage_option |
  (nmpd ::= mfuo.relatering_product_definition) )) >= 1)) )) = 0;
END_RULE;
(*
```

Argument definitions:

**product\_definition:** the set of all instances of **product\_definition** entities.

**make\_from\_usage\_option:** the set of all instances of **make\_from\_usage\_option** entities.

Formal propositions:

**WR1:** For each instance of **product\_definition** that is not designated as a material through a reference by the **definitions** attribute of the **material\_designation**, there shall be at least one instance of **make\_from\_usage\_option** in which the non-material **product\_definition** is the **relatering\_product\_definition**.

### 5.2.4.20 part\_requires\_project\_order

The **part\_requires\_project\_order** rule specifies that each instance of **product\_definition** shall be referenced by exactly one instance of **applied\_action\_assignment** that assigns a project order to the part.

EXPRESS specification:

```
*)
RULE part_requires_project_order FOR (product_definition,
  applied_action_assignment);
WHERE
wr1: SIZEOF(QUERY ( pd <* product_definition |
  ((pd.frame_of_reference.name='part definition') AND
  (NOT (SIZEOF(QUERY ( fbppaa <* applied_action_assignment |
  ((pd.formation IN fbppaa.items) AND
  (fbppaa.assigned_action.name = 'project order')) )) =1 )) )) =0;
END_RULE;
(*
```

Argument definitions:

**product\_definition:** the set of all instances of **product\_definition** entities.

**applied\_action\_assignment:** the set of all instances of **applied\_action\_assignment** entities.

Formal propositions:

**WR1:** For each instance of **product\_definition**, there shall be exactly one instance of **applied\_action\_assignment** that contains the instance of **product\_definition\_formation** in its set of **items** and references an **action** with a name of 'project order'.

### 5.2.4.21 part\_to\_approval

The **part\_to\_approval** rule specifies that each instance of **product\_definition\_formation** shall be referenced by at most one instance of **applied\_approval\_assignment**.

EXPRESS specification:

```
*)
RULE part_to_approval FOR (product_definition_formation,
  applied_approval_assignment);
WHERE
WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
  NOT (SIZEOF (QUERY (fbppa <* applied_approval_assignment |
  pdf IN fbppa.items )) <= 1 ))) = 0;
END_RULE;
(*
```

ISO 10303-223:2008 (E)

Argument definitions:

**product\_definition\_formation:** the set of all instances of **product\_definition\_formation** entities.

**applied\_approval\_assignment:** the set of all instances of **applied\_approval\_assignment** entities.

Formal propositions:

**WR1:** For each instance of **product\_definition\_formation**, there shall be exactly one instance of **applied\_approval\_assignment** that contains the instance of **product\_definition\_formation** in its set of items.

#### 5.2.4.22 product\_requires\_version

The **product\_requires\_version** rule specifies that each instance of **product** shall be referenced by at least one instance of **product\_definition\_formation**. This rule enforces the requirement for every product to have one or more versions.

EXPRESS specification:

\*)

```
RULE product_requires_version FOR (product, product_definition_formation);
WHERE
  WR1: SIZEOF (QUERY (prod <* product |
    NOT (SIZEOF (QUERY (pdf <* product_definition_formation |
      prod ::= pdf.of_product )) >= 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

**product:** the set of all instances of **product** entities.

**product\_definition\_formation:** the set of all instances of **product\_definition\_formation** entities.

Formal propositions:

**WR1:** For each instance of **product**, there shall be one or more instances of **product\_definition\_formation** that contains an **of\_product** attribute value equal to that instance of **product**.

#### 5.2.4.23 part\_requires\_security\_classification

The **part\_requires\_security\_classification** rule specifies that each instance of **product\_definition\_formation** shall be referenced by exactly one instance of **applied\_security\_classification\_assignment**. This rule enforces the requirement for every version of a design to have a security classification.

EXPRESS specification:

```

*)
  RULE part_requires_security_classification FOR (
      product_definition,
      applied_security_classification_assignment);

  WHERE
  wr1: SIZEOF(QUERY ( pd <* product_definition |
      ((pd.frame_of_reference.name='part definition') AND

      (NOT (SIZEOF(QUERY
      ( fbppsc <* applied_security_classification_assignment |
      (pd.formation IN fbppsc.items) )) =1 )) )) )=0;

  END_RULE; -- part_requires_security_classification
(*

```

Argument definitions:

**product\_definition:** the set of all instances of **product\_definition** entities.

**applied\_security\_classification\_assignment:** the set of all instances of **applied\_security\_classification\_assignment** entities.

Formal propositions:

**WR1:** For each instance of **product\_definition**, there shall be exactly one instance of **applied\_security\_classification\_assignment** that contains the instance of **product\_definition\_formation** in its set of **items**.

### 5.2.4.24 project\_order\_requires\_approval

The **project\_order\_requires\_approval** rule specifies that each instance of **directed\_action** that references an **action** with a **name** of 'project order' shall be referenced by exactly one instance of **applied\_approval\_assignment**. This rule enforces the requirement for every project order to have an approval.

ISO 10303-223:2008 (E)

EXPRESS specification:

```
*)
RULE project_order_requires_approval FOR
  (directed_action, applied_approval_assignment);
WHERE
  WR1: SIZEOF (QUERY (po <* QUERY (da <* directed_action |
    da.name = 'project order') |
    NOT (SIZEOF (QUERY (fbppapp <* applied_approval_assignment |
      po IN fbppapp.items )) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

**directed\_action:** the set of all instances of **directed\_action** entities.

**applied\_approval\_assignment:** the set of all instances of **applied\_approval\_assignment** entities.

Formal propositions:

**WR1:** For each instance of **directed\_action** there shall be exactly one instance of **applied\_approval\_assignment** that contains the instance of **directed\_action** in its set of **items**.

### 5.2.4.25 project\_order\_tracking\_relationships

The **project\_order\_tracking\_relationships** rule specifies the relationship between a project order and the other types of orders within the process planning preparation. Each order shall be carried out under a project order, and an individual project order shall be defined for tracking at most one of each type of order within the project. The types of orders defined for this part of ISO 10303 are shop work order, resource acquisition order, digital technical data package work order, and pedigree creation order.

EXPRESS specification:

```
*)
RULE project_order_tracking_relationships FOR (directed_action,
  action_relationship);
WHERE
  WR1: SIZEOF (QUERY (da <* directed_action |
    (da.name IN ['shop work order', 'resource acquisition order',
      'digital technical data package work order',
      'pedigree creation order']) AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.related_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
        TYPEOF (ar.relating_action)) AND
      (ar.relating_action.name = 'project order')))) = 1))) = 0;
  WR2: SIZEOF (QUERY (da <* directed_action |
    (da.name = 'project order') AND
    NOT (SIZEOF (QUERY (ar <* action_relationship |
```



```

      (da :=: ar.relating_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'shop work order')))) <= 1))) = 0;
WR3: SIZEOF (QUERY (da <* directed_action |
      (da.name = 'project order') AND
      NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relating_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'resource acquisition order'))))
      <= 1))) = 0;
WR4: SIZEOF (QUERY (da <* directed_action |
      (da.name = 'project order') AND
      NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relating_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.related_action)) AND
      (ar.related_action.name =
      'digital technical data package work order')))) <= 1))) = 0;
WR5: SIZEOF (QUERY (da <* directed_action |
      (da.name = 'project order') AND
      NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relating_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'pedigree creation order'))))
      <= 1))) = 0;
WR6: SIZEOF (QUERY (da <* directed_action |
      (da.name = 'customer order') AND
      NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.related_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.relating_action)) AND
      (ar.relating_action.name = 'project order')))) >= 1))) = 0;
WR7: SIZEOF (QUERY (da <* directed_action |
      (da.name = 'project order' ) AND
      NOT (SIZEOF (QUERY (ar <* action_relationship |
      (da :=: ar.relating_action) AND
      (('CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN
      TYPEOF (ar.related_action)) AND
      (ar.related_action.name = 'customer order'))))
      <= 1))) = 0;
END_RULE;
(*

```

### Argument definitions:

**directed\_action:** the set of all instances of **directed\_action** entities.

**action\_relationship:** the set of all instances of **action\_relationship** entities.

Formal propositions:

**WR1:** For each instance of **directed\_action** with a **name** of either 'shop work order', 'resource acquisition order', 'digital technical data package work order', or 'pedigree creation order', there shall be exactly one instance of **action\_relationship** that references that **directed\_action** as the **related\_action**, and references an instance of **directed\_action** with a **name** of 'project order' as the **relating\_action**.

**WR2:** For each instance of **directed\_action** with a **name** of 'project order', there shall be at most one instance of **action\_relationship** that references that **directed\_action** as the **relating\_action**, and references an instance of **directed\_action** with a **name** of 'shop work order'.

**WR3:** For each instance of **directed\_action** with a **name** of 'project order', there shall be at most one instance of **action\_relationship** that references that **directed\_action** as the **relating\_action**, and references an instance of **directed\_action** with a **name** of 'resource acquisition order'.

**WR4:** For each instance of **directed\_action** with a **name** of 'project order', there shall be at most one instance of **action\_relationship** that references that **directed\_action** as the **relating\_action**, and references an instance of **directed\_action** with a **name** of 'digital technical data package work order'.

**WR5:** For each instance of **directed\_action** with a **name** of 'project order', there shall be at most one instance of **action\_relationship** that references that **directed\_action** as the **relating\_action**, and references an instance of **directed\_action** with a **name** of 'pedigree creation order'.

**WR6:** For each instance of **directed\_action** with a **name** of 'customer order' there shall be at least one instance of **action\_relationship** that references that **directed\_action** as the **related\_action**, and references an instance of **directed\_action** with a **name** of 'project order' as the **relating\_action**.

**WR7:** For each instance of **directed\_action** with a **name** of 'project order', there shall be at most one instance of **action\_relationship** that references that **directed\_action** as the **relating\_action**, and references an instance of **directed\_action** with a **name** of 'customer order'.

#### **5.2.4.26 restrict\_approval\_status**

The **restrict\_approval\_status** rule specifies that the only values of **approval\_status** permitted shall be 'approved', 'not\_yet\_approved', 'disapproved', or 'withdrawn'.

EXPRESS specification:

```
*)
RULE restrict_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
    NOT (ast.name IN
      ['approved', 'not yet approved', 'disapproved', 'withdrawn']))) = 0;
END_RULE;
(*
```

Argument definitions:

**approval\_status:** the set of all instances of **approval** entities.

Formal propositions:

**WR1:** For each instance of **approval**, the value of the **approval\_status** attribute shall be either 'approved', 'not yet approved', 'disapproved', or 'withdrawn'.

Attribute value definitions:

**approved:** specifies that the required authorizations have been obtained for a particular role of approval for a piece of product data.

**not\_yet\_approved:** specifies that the required authorizations are being awaited for a particular role of approval for a piece of product data.

**disapproved:** specifies that the required authorizations have been denied for a particular role of approval for a piece of product data.

**withdrawn:** specifies that the required authorizations have been revoked for a particular role of approval for a piece of product data.

### 5.2.4.27 restrict\_externally\_defined\_feature\_definition

The **restrict\_externally\_defined\_feature\_definition** is a global rule that restricts the use of **externally\_defined\_feature\_definition**.

EXPRESS specification

\*)

```

RULE restrict_externally_defined_feature_definition
  FOR (EXTERNALLY_DEFINED_FEATURE_DEFINITION);
WHERE
  wr1:((SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
    (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
      'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
      ('CAST_PARTS_SCHEMA.DOCUMENT_FILE'
      IN TYPEOF(adr.assigned_document)))) >=1)))) =0) OR
    (SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
      (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
        'CAST_PARTS_SCHEMA'+
        '.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS') |
        ('CAST_PARTS_SCHEMA.DOCUMENT_FILE'
        IN TYPEOF(adr\document_usage_constraint_assignment.
          assigned_document_usage.source))))
          >=1)))) =0)));
END_RULE;
```

ISO 10303-223:2008 (E)

(\*

Formal propositions:

**WR1:** The **externally\_defined\_feature\_definition** shall be in the set of **items** of more than one **applied\_document\_reference** or exactly one **applied\_document\_usage\_constraint\_assignment** that defines the **document\_file** containing the feature specification.

### 5.2.4.28 restrict\_security\_classification\_level

The **restrict\_security\_classification\_level** rule specifies the permitted levels of security. This rule enforces the requirement for the levels of security to be "unclassified", "classified", "proprietary", "confidential", "secret", or "top\_secret".

EXPRESS specification

```
*)
RULE restrict_security_classification_level FOR
  (security_classification_level);
WHERE
  WR1: SIZEOF (QUERY (scl <* security_classification_level |
    NOT (scl.name IN
      ['unclassified', 'classified', 'proprietary', 'confidential', 'secret',
      'top_secret']))) = 0;
END_RULE;
(*
```

Argument definitions:

**security\_classification\_level:** identifies the set of all instances of **security\_classification\_level** entities.

Formal propositions:

**WR1:** For each instance of **security\_classification\_level**, the **name** attribute shall contain a value of "unclassified", "classified", "proprietary", "confidential", "secret", or "top\_secret".

Attribute value definitions:

**unclassified:** identifies the classification level for which no security is necessary.

**classified:** identifies the classification level for which security is necessary, but the classification details are not given.

**proprietary:** identifies the classification level for which the disclosure of information about the part or the design of the part would risk an organization's market or competitive advantage.

**confidential:** identifies the classification level for which the disclosure of information about the part or the design of the part would cause damage to national or organizational security.

**secret:** identifies the classification level for which the disclosure of information about the part or the design of the part would cause serious damage to national or organizational security.

**top\_secret:** identifies the classification level for which the disclosure of information about the part or the design of the part would cause exceptionally grave damage to national or organizational security.

### 5.2.4.29 restrict\_name\_for\_known\_source

#### EXPRESS specification

```
*)
RULE restrict_name_for_known_source FOR (known_source);

WHERE
  wr1: SIZEOF(QUERY ( ks <* known_source |
    ((ks.name :<>: 'ISO 13584 library') AND
     (ks.name :<>: 'ISO 10303 part')))) = 0;

END_RULE; -- restrict_name_for_known_source
(*)
```

### 5.2.4.30 subtype\_mandatory\_characterized\_object

The **subtype\_mandatory\_characterized\_object** rule specifies the permitted usage of the **characterized\_object**. The **characterized\_object** entity shall be limited to its use in the definition of a **feature\_definition**, **feature\_component\_definition**, or **ordered\_part**.

#### EXPRESS specification:

```
*)

RULE subtype_mandatory_characterized_object FOR (characterized_object);
WHERE
  wr1: ((SIZEOF(QUERY ( csa <* characterized_object |
    (NOT (SIZEOF([
      'CAST_PARTS_SCHEMA.DOCUMENT_FILE',
      'CAST_PARTS_SCHEMA.FEATURE_DEFINITION',
      'CAST_PARTS_SCHEMA.CASTING_FEATURE_DEFINITION',
      'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION',
      'CAST_PARTS_SCHEMA.ORDERED_PART'] *
      TYPEOF(csa)) = 1))
    AND
    (NOT(SIZEOF(QUERY ( pd <* USEDIN(csa,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('CAST_PARTS_SCHEMA.MATERIAL_PROPERTY'
      IN TYPEOF(pd)) )) = 1))
      )) = 0) );
  END_RULE; -- subtype_mandatory_characterized_object
(*)
```

Argument definitions:

**characterized\_object:** the set of all instances of **characterized\_object** entities.

Formal propositions:

**WR1:** Each instance of **characterized\_object** shall be either a **feature\_definition**, **feature\_component\_definition**, or **ordered\_part**, or referenced by the **definition** attribute of **material\_property**.

### 5.2.4.31 transition\_feature\_life\_cycle

The **transition\_feature\_life\_cycle** rule specifies that each instance of **transition\_feature** shall be defined for the manufacturing planning stage of the part on which it is specified.

EXPRESS specification:

```
*)  
RULE transition_feature_life_cycle FOR  
  (transition_feature);  
WHERE  
  WR1: SIZEOF (QUERY (tf <* transition_feature |  
    NOT (tf.of_shape.definition.frame_of_reference.life_cycle_stage =  
      'manufacturing planning')) = 0;  
END_RULE;  
(*
```

Argument definitions:

**transition\_feature:** the set of all instances of **transition\_feature** entities.

Formal propositions:

**WR1:** For each instance of **transition\_feature**, the **life\_cycle\_stage** of the **product\_definition** for which it is defined has a value of 'manufacturing planning'.

### 5.2.4.32 transition\_feature\_on\_part\_boundary

The **transition\_feature\_on\_part\_boundary** rule specifies that each instance of **transition\_feature** shall lie on the boundary of the part for which it is defined.

EXPRESS specification:

```

*)
RULE transition_feature_on_part_boundary FOR
  (transition_feature);
WHERE
  WR1: SIZEOF (QUERY (tf <* transition_feature |
    NOT (tf.product_definitional))) = 0;
END_RULE;
(*

```

Argument definitions:

**transition\_feature:** the set of all instances of **transition\_feature** entities.

Formal propositions:

**WR1:** For each instance of **transition\_feature**, **product\_definitional** shall have a value of true.

## 5.2.5 Forged parts schema functions

### 5.2.5.1 get\_property\_definition\_shape\_aspect

The **get\_property\_definition\_shape\_aspect** function returns for any **characterized\_definition** the set of **shape\_aspect** that refer to the **characterized\_definition** through a **property\_definition**.

EXPRESS specification:

```

*)

FUNCTION get_property_definition_shape_aspect(
  c_def_instance: characterized_definition
): SET OF shape_aspect;

LOCAL
  pdr_set : SET OF shape_aspect := [];
  pd_set  : SET OF property_definition := [];
END_LOCAL;
pd_set := bag_to_set(USEDIN(c_def_instance,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION'));
IF SIZEOF(pd_set) < 1 THEN
  RETURN(pdr_set);
END_IF;
REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
  pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE'));
END_REPEAT;
RETURN(pdr_set);

END_FUNCTION;
(*

```

ISO 10303-223:2008 (E)

Argument definitions:

**c\_def\_instance**: (input) the candidate **characterized\_definition** to be checked.

### 5.2.5.2 get\_action\_property\_representation

The **get\_action\_property\_representation** function returns for any **characterized\_action\_definition** the set of **action\_property\_representation** that refer to the **characterized\_action\_definition** through a **action\_property\_representation**.

EXPRESS specification:

\*)

```
FUNCTION get_action_property_representation(  
    c_def_instance: characterized_action_definition  
): SET OF action_property_representation;  
  
LOCAL  
    pdr_set : SET OF action_property_representation := [];  
    pd_set  : SET OF action_property := [];  
END_LOCAL;  
pd_set := bag_to_set(USEDIN(c_def_instance,  
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION'));  
IF SIZEOF(pd_set) < 1 THEN  
    RETURN(pdr_set);  
END_IF;  
REPEAT i := 1 TO HIINDEX(pd_set) BY 1;  
    pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],  
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY'));  
END_REPEAT;  
RETURN(pdr_set);  
  
END_FUNCTION;  
(*
```

Argument definitions:

**c\_def\_instance**: (input) the candidate **characterized\_action\_definition** to be checked.

### 5.2.5.3 get\_resource\_property\_representation

The **get\_action\_property\_representation** function returns for any **characterized\_resource\_definition** the set of **resource\_property\_representation** that refer to the **characterized\_resource\_definition** through a **resource\_property\_representation**.



EXPRESS specification:

\*)

```

FUNCTION get_resource_property_representation(
    c_def_instance: characterized_resource_definition
): SET OF resource_property_representation;

LOCAL
    pdr_set : SET OF resource_property_representation:= [];
    pd_set  : SET OF resource_property := [];
END_LOCAL;
pd_set := bag_to_set(USEDIN(c_def_instance,
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE'));
IF SIZEOF(pd_set) < 1 THEN
    RETURN(pdr_set);
END_IF;
REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
    pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY'));
END_REPEAT;
RETURN(pdr_set);

END_FUNCTION;

END_SCHEMA; -- forged_parts_schema
(*

```

Argument definitions:

**c\_def\_instance:** (input) the candidate **characterized\_resource\_definition** to be checked.

\*)

```

END_SCHEMA; -- cast_parts_schema
(*

```

## 6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation method(s) supported, and the relevant requirements of the normative references.

An implementation shall support at least one of the following implementation methods:

— ISO 10303-21.

Requirements with respect to implementation methods-specific requirements are specified in Annex C. A usage guide to aid consistent implementations is specified in Annex J.

The Protocol Implementation Conformance Statement (PICS) proforma lists the options or the combinations of options that may be included in the implementation. The PICS proforma is provided in Annex D.

This part of ISO 10303 provides for a number of options that may be supported by an implementation. These options have been grouped into the following conformance classes: Eight conformance classes are defined. Conformance to this part of ISO 10303 requires, a minimum, conformance to class 1 and may be selected by an implementation. Support for a particular conformance class requires support of all the options specified in that class.

The conformance classes are specified using combinations of UoFs as given in Table 1. Conformance class 1 is required as a minimum conformance, classes 2 to 8 conform to class 1 as well as their own conformance. This part of ISO 10303 uses three AICs. These AIC define boundary representation geometry, machining features, and geometric tolerances which are used by all conformance classes.

Conformance to a particular class requires that all AIM elements defined as part of that class be supported. Table 2 defines the classes to which each AIM element belongs.

The conformance classes are characterized as follows:

- Class 1: Minimum for all conformance classes;
- Class 2: Customer to metal caster;
- Class 3: Metal caster to inspection;
- Class 4: Metal caster to simulation;
- Class 5: Metal caster to tooling shop;
- Class 6: Metal caster to other operations;
- Class 7: Customer to simulation;
- Class 8: Quality assurance;
- Class 9: Casting process planning.

**Table 1 — Conformance class UoFs**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
<b>Casting feature UoF</b>			X	X	X	X		X	X
Casting_design_feature			X	X	X	X		X	X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Casting_insert			X	X	X	X		X	X
Chaplet			X	X	X	X		X	X
Chaplet_pad			X	X	X	X		X	X
Chill			X	X	X	X		X	X
Choke			X	X	X	X		X	X
Connection_transition			X	X	X	X		X	X
Core			X	X	X	X		X	X
Core_box_tooling			X	X	X	X		X	X
Core_master			X	X	X	X		X	X
Core_print			X	X	X	X		X	X
Ejector_box			X	X	X	X		X	X
Ejector_design_feature			X	X	X	X		X	X
Ejector_pins			X	X	X	X		X	X
Ejector_plate			X	X	X	X		X	X
Ejector_system			X	X	X	X		X	X
Feeder			X	X	X	X		X	X
Filter			X	X	X	X		X	X
Fixturing_tab			X	X	X	X		X	X
Flask			X	X	X	X		X	X
Flaskless			X	X	X	X		X	X
Gating_design_feature			X	X	X	X		X	X
Gating_system			X	X	X	X		X	X
General_rib			X	X	X	X		X	X
Ingate			X	X	X	X		X	X
Ingate_contacts			X	X	X	X		X	X
Item_size			X	X	X	X		X	X
Loose_piece			X	X	X	X		X	X
Master_pattern			X	X	X	X		X	X
Mould_box			X	X	X	X		X	X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Mould_locks			X	X	X	X		X	X
Pattern_plate			X	X	X	X		X	X
Pin_center			X	X	X	X		X	X
Pin_tips			X	X	X	X		X	X
Planar_rib			X	X	X	X		X	X
Production_core_box			X	X	X	X		X	X
Production_pattern_definition			X	X	X	X		X	X
Reset_or_return_pins			X	X	X	X		X	X
Rib			X	X	X	X		X	X
Riser_contact			X	X	X	X		X	X
Risers			X	X	X	X		X	X
Runner			X	X	X	X		X	X
Sand_cast_design_feature			X	X	X	X		X	X
Sand_casting_tooling			X	X	X	X		X	X
Sand_mould			X	X	X	X		X	X
Sprue			X	X	X	X		X	X
Stop_pin			X	X	X	X		X	X
Vent			X	X	X	X		X	X
Well			X	X	X	X		X	X
<b>Casting_prerequisite UoF</b>									
Casting_service_data		X	X	X	X		X	X	
Corrosion_service		X	X	X	X		X	X	
Customer_casting_requirement		X	X	X	X		X	X	
Elevated_temperature_service		X	X	X	X		X	X	
Finishing_or_machining_operations		X	X	X	X		X	X	
Heat_treat_requirement		X	X	X	X		X	X	
Inspection_or_test_requirement		X	X	X	X		X	X	
Inspection_plan		X	X	X	X		X	X	
Machining_allowance		X	X	X	X		X	X	

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Mechanical_property_requirements		X	X	X	X		X	X	
Mechanical_stress		X	X	X	X		X	X	
Mismatch_tolerance		X	X	X	X		X	X	
Pattern_equipment_available		X	X	X	X		X	X	
Payment_and_shipping		X	X	X	X		X	X	
Process_requirement		X	X	X	X		X	X	
Reporting_requirement		X	X	X	X		X	X	
Special_inspection_requirement		X	X	X	X		X	X	
Tolerance_requirement		X	X	X	X		X	X	
<b>Design_exception UoF</b>									
Design_exception_notice		X							X
Engineering_change_order		X							X
Engineering_change_proposal		X							X
Request_for_clarification		X							X
<b>Die_mould_features UoF</b>									
Cooling_port			X	X	X	X	X	X	X
Die_design_feature			X	X	X	X	X	X	X
Die_cast_master			X	X	X	X	X	X	X
Die_cast_tooling			X	X	X	X	X	X	X
Die_clamping			X	X	X	X	X	X	X
Die_core			X	X	X	X	X	X	X
Die_master			X	X	X	X	X	X	X
Die_mould_tooling			X	X	X	X	X	X	X
Die_mould_vent			X	X	X	X	X	X	X
Production_die_cast_mould			X	X	X	X	X	X	X
Production_die_mould			X	X	X	X	X	X	X
Production_tool			X	X	X	X	X	X	X
Secondary_tooling			X	X	X	X	X	X	X
Shot_sleeve			X	X	X	X	X	X	X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Slide			X	X	X	X	X	X	X
<b>Investment_casting_features UoF</b>									
Investment_casting_master			X	X	X	X	X	X	X
Investment_casting_tooling			X	X	X	X	X	X	X
Investment_casting_tooling_definition			X	X	X	X	X	X	X
Investment_design_feature			X	X	X	X	X	X	X
Investment_mould			X	X	X	X	X	X	X
Lost_foam_casting_die			X	X	X	X	X	X	X
Production_investment_cast_mould			X	X	X	X	X	X	X
Sprue_and_runner_mould			X	X	X	X	X	X	X
<b>Library_reference UoF</b>									
Bsu	X								
Class_bsu	X								
Externally_defined_representation	X								
Library_part_assignment	X								
Property_bsu	X								
Property_value	X								
Supplier_bsu	X								
Bsu	X								
<b>Manufacturing_casting_resources UoF</b>									
Bankout_frame					X			X	X
Controller					X			X	X
Core_bench					X			X	X
Core_box_vent					X			X	X
Core_conveyer_system					X			X	X
Core_equipment					X			X	X
Core_machine					X			X	X
Core_setter					X			X	X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Dryer					X			X	X
Ejector_equipment					X			X	X
Equipment					X			X	X
Equipment_size					X			X	X
Finishing					X			X	X
Generic_casting_resource					X			X	X
In_facility_location					X			X	X
Machine_mounting					X			X	X
Melting_equipment					X			X	X
Moulding_equipment					X			X	X
Resource_with_material					X			X	X
Resource_with_representation					X			X	X
Shakeout					X			X	X
<b>Manufacturing_feature UoF</b>									
Compound_feature	X								
Compound_feature_element	X								
Compound_feature_relationship	X								
Feature	X								
Machining_feature	X								
Manufacturing_design_feature	X								
Manufacturing_feature	X								
Manufacturing_feature_group	X								
Transition_feature	X								
<b>Manufacturing_part_properties UoF</b>									
Casting_weight_requirement		X			X			X	
Composition_element								X	
Data_curve								X	
Descriptive_parameter	X								

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Hardness	X								
Material	X								
Material_composition								X	
Material_property	X								
Material_structure								X	
Material_structure_element								X	
Numeric_parameter	X								
Numeric_parameter_with_tolerance	X								
Numeric_range	X								
Part_property	X								
Process_property	X								
Property	X								
Property_parameter	X								
Property_relationship	X								
Surface_property	X								
<b>Manufacturing process - requirement documents UoF</b>									
Customer_design_deficiency_report		X							
Customer_tool_design_report					X				
Design_reference		X	X	X	X	X	X	X	X
Digital_file	X								
Document_assignment	X								
Document_file	X								
Document_file_properties	X								
Document_file_relationship	X								
Executable	X								
External_file_identification	X								
External_schema_definition	X								
Hardcopy	X								



**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Illustration		X	X	X	X	X	X	X	X
Part_dimensioning_standard	X								
Product_quality_report								X	
Proof_report								X	
Quality_acceptance_report								X	
Simulation_exception_report							X	X	
Simulation_report							X		
Special_instruction									X
Specification	X								
Specification_usage_constraint	X								
Supplemental_document	X								
View_reference		X	X	X	X	X	X	X	X
<b>Manufacturing_process_- control_documentation UoF</b>									
Customer_order		X							X
Digital_technical_data_package_work_order		X							X
Ordered_part		X							X
Pedigree_creation_order		X							X
Project_order		X							X
Request_for_quotation		X							X
Resource_acquisition_order		X							X
Shop_work_order		X							X
<b>Measurement_limitations UoF</b>									
Angular_dimension_tolerance	X								
Angular_size_dimension_tolerance	X								
Common_datum	X								
Curved_dimension_tolerance	X								
Datum	X								
Datum_feature	X								

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Datum_target	X								
Datum_target_set	X								
Diameter_dimension_tolerance	X								
Dimensional_tolerance	X								
Distance_along_curve_tolerance	X								
Externally_defined_size_dimension	X								
Geometric_tolerance	X								
Geometric_tolerance_precedence_relationship	X								
Height_dimension	X								
Length_dimension	X								
Limits_and_fits	X								
Location_dimension_tolerance	X								
Location_tolerance	X								
Material_condition_modifier	X								
Placed_target	X								
Plus_minus_value	X								
Projection	X								
Radial_dimension_tolerance	X								
Size_tolerance	X								
Target_area	X								
Target_circle	X								
Target_line	X								
Target_point	X								
Target_rectangle	X								
Thickness_tolerance	X								
Tolerance_limit	X								
Tolerance_range	X								
Tolerance_value	X								
Tolerance_zone	X								

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Tolerance_zone_definition	X								
Width_dimension	X								
<b>Part_administration_data UoF</b>									
Address	X								
Approval	X								
Date	X								
Organization	X								
Person	X								
Person_in_organization	X								
Status_authority	X								
<b>Part_model UoF</b>									
Cast_part			X	X	X	X	X	X	X
Cast_part_with_rigging			X	X	X	X	X	X	X
Design_part	X								
Master_sample								X	
Part_relationship			X	X	X	X	X	X	X
Part_version	X								
Red_lined_part		X							
Tooling			X	X	X	X	X	X	X
<b>Process_activities UoF</b>									
Activity									X
Allowed_time									X
Alternate_activity									X
Ancillary_activity									X
Ancillary_setup									X
Assembly									X
Bench_inspection									X
Casting_activity			X	X	X	X	X		X
Customer_simulation									X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Dimensional_measurement_inspection									X
General_foundry_process									X
Equipment_process									X
Equipment_setup									X
Fabrication									X
Heat_treat									X
Impregnation_activity									X
Inspection_activity									X
Machining_activity									X
Manufacturing_process									X
Metalcaster_simulation									X
Non_casting_operations									X
Non_equipment_process									X
Non_permanent_moulding_process			X	X	X	X	X		X
Performance_rate									X
Permanent_moulding_process			X	X	X	X	X		X
Production_rate									X
Production_welding									X
Setup_activity									X
Simulation_process									X
Surface_finish									X
Tooling_process									X
<b>Process_plan UoF</b>									
Alternate_process_plan									X
Continuous_process									X
Process_plan			X	X	X	X	X		X
Process_plan_security									X
Process_plan_version									X
Revision									X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
<b>Quality_control UoF</b>								X	
Activity_execution_result								X	
Activity_result_record								X	
Casting_per_order								X	
Casting_verification								X	
Checking_aid_tool								X	
Composition_element_record								X	
Composition_inspection								X	
Core_assembly								X	
Equipment_setting_record								X	
Equipment_usage								X	
First_article								X	
Heat								X	
Inspection_or_test_result								X	
Lot								X	
Material_usage								X	
Material_usage_record								X	
Other_inspection								X	
Process_parameter_record								X	
Property_inspection								X	
Sampled_set								X	
Shape_dimensions								X	
Tolerance_inspection								X	
Tool_verification								X	
<b>Requisitions UoF</b>									
Die_mould_requisition									X
Indirect_stock_requisition									X
Material_requisition									X
Pattern_tooling_requisition									X

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Requisition									X
<b>Shape_representation - for casting and machining UoF</b>									
Base_shape	X								
Casting_envelope_requirement		X			X				
Curve_point				X					
Direction_element	X								
Drafted_surface	X								
Edge_shape_element	X								
Face_shape_element	X								
Face_shape_element_relationship	X								
Geometric_model	X								
Location_element	X								
Model_element	X								
Object_element_shape_representation	X								
Orientation	X								
Part_shape	X								
Parting_surface	X								
Path_element	X								
Planar_element	X								
Shape_aspect	X								
Shape_aspect_representation	X								
Shape_element	X								
Shape_representation_type	X								
<b>Simulation UoF</b>									
Boundary_condition				X			X		
Condition_or_assumption				X			X		
Defect_prediction				X			X		
Integration_interval				X			X		

**Table 1 — Conformance class UoFs (continued)**

Unit of Functionality	Class								
	1	2	3	4	5	6	7	8	9
Meshing_condition				X			X		
Simulated_property				X			X		
Simulation_input				X			X		
Simulation_input_region				X			X		
Simulation_output				X			X		
Simulation_output_region				X			X		
Simulation_result				X			X		
Simulation_run				X			X		
Simulation_run_relationship				X			X		
Simulation_software				X			X		
Simulation_tooling_report				X			X		
Simulation_unit				X			X		
Simulation_unit_state				X			X		

**Table 2 — Conformance class elements**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Action	X								
Action_assignment	X								
Action_directive	X								
Action_method	X								
Action_method_assignment	X								
Action_method_relationship	X								
Action_method_role	X								
Action_method_with_associated_documents	X								
Action_property	X								
Action_property_relationship	X								
Action_property_representation	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Action_relationship	X								
Action_request_assignment	X								
Action_request_solution	X								
Action_resource	X								
Action_resource_relationship	X								
Action_resource_requirement	X								
Action_resource_type	X								
Action_status	X								
Activity_execution_result	X							X	
Activity_result_record	X							X	
Address	X								
Advanced_brep_shape_representation (AIC 514)	X								
Advanced_face (AIC 511)	X								
Allowed_time									X
Alternate_action_method_relationship	X								
Alternate_plan_relationship									X
Ancillary_activity									X
Ancillary_setup									X
Angular_location	X								
Angular_size	X								
Angularity_tolerance (AIC 519)	X								
Apex	X								
Application_context	X								
Application_context_element	X								
Application_protocol_definition	X								
Applied_action_assignment	X								
Applied_action_method_assignment	X								
Applied_action_request_assignment	X								
Applied_approval_assignment	X								



**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Applied_area (AIC 522)	X								
Applied_classification_assignment	X								
Applied_date_and_time_assignment	X								
Applied_date_assignment	X								
Applied_document_reference	X								
Applied_document_usage_constraint_assignment	X								
Applied_effectivity_assignment	X								
Applied_external_identification_assignment	X								
Applied_group_assignment	X								
Applied_identification_assignment	X								
Applied_library_assignment	X								
Applied_organization_assignment	X								
Applied_person_and_organization_assignment	X								
Applied_security_classification_assignment	X								
Applied_time_assignment	X								
Approval	X								
Approval_assignment	X								
Approval_date_time	X								
Approval_person_organization	X								
Approval_role	X								
Approval_status	X								
Assembly									X
Assembly_component_usage	X								
Axis1_placement	X								
Axis2_placement_2d	X								
Axis2_placement_3d	X								
B_spline_curve	X								
B_spline_curve_with_knots	X								
B_spline_surface	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
B_spline_surface_with_knots	X								
Bankout_frame					X			X	X
Bench_inspection									X
Bezier_curve	X								
Bezier_surface	X								
Boss (AIC 522)	X								
Boss_top (AIC 522)	X								
Boundary_condition				X			X		
Bounded_curve	X								
Bounded_surface	X								
Brep_with_voids	X								
Calendar_date	X								
Cartesian_point	X								
Cartesian_transformation_operator	X								
Cartesian_transformation_operator_3d	X								
Cast_part			X	X	X	X	X	X	X
Cast_part_with_rigging			X	X	X	X	X	X	X
Casting_activity			X	X	X	X	X		X
Casting_design_feature			X	X	X	X		X	X
Casting_envelope_requirement		X			X				
Casting_equipment					X			X	X
Casting_equipment_process									X
Casting_equipment_setup									X
Casting_feature_definition			X	X	X	X		X	X
Casting_feature_size			X	X	X	X		X	X
Casting_insert			X	X	X	X		X	X
Casting_instanced_feature			X	X	X	X		X	X
Casting_method								X	
Casting_per_order									X

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Casting_process									X
Casting_product_definition			X	X	X	X	X	X	X
Casting_service_data		X	X	X	X		X	X	
Casting_verification									X
Centre_of_symmetry	X								
Chamfer (AIC 522)	X								
Chamfer_offset (AIC 522)	X								
Chaplet			X	X	X	X		X	X
Chaplet_pad			X	X	X	X		X	X
Characterized_object	X								
Checking_aid_tool									X
Chill			X	X	X	X		X	X
Choke			X	X	X	X		X	X
Circle	X								
Circular_closed_profile (AIC 522)	X								
Circular_pattern (AIC 522)	X								
Circular_runout_tolerance (AIC 519)	X								
Class	X								
Classification_assignment	X								
Classification_role	X								
Closed_path_profile (AIC 522)	X								
Closed_shell	X								
Coaxiality_tolerance (AIC 519)	X								
Common_datum (AIC 519)	X								
Composite_curve	X								
Composite_curve_on_surface	X								
Composite_curve_segment	X								
Composite_hole (AIC 522)	X								
Composite_shape_aspect	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Composition_element								X	
Composition_element_record								X	
Composition_inspection								X	
Compound_feature (AIC 522)	X								
Compound_representation_item	X								
Concentricity_tolerance (AIC 519)	X								
Condition_or_assumption				X			X		
Conic	X								
Conical_surface	X								
Connected_face_set	X								
Context_dependent_unit	X								
Continuous_process_relationship									X
Controller					X			X	X
Conversion_based_unit	X								
Cooling_port			X	X	X	X	X	X	X
Coordinated_universal_time_offset	X								
Core			X	X	X	X		X	X
Core_assembly								X	
Core_bench					X			X	X
Core_box_tooling			X	X	X	X		X	X
Core_box_vent					X			X	X
Core_conveyer_system					X			X	X
Core_equipment					X			X	X
Core_machine					X			X	X
Core_master			X	X	X	X		X	X
Core_print			X	X	X	X		X	X
Core_setter					X			X	X
Corrosion_service		X	X	X	X		X	X	
Curve	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Customer_casting_requirement		X	X	X	X		X	X	
Customer_simulation									X
Cylindrical_surface	X								
Cylindricity_tolerance	X								
Data_curve								X	
Data_environment	X								
Date	X								
Date_and_time	X								
Date_and_time_assignment	X								
Date_assignment	X								
Date_role	X								
Date_time_role	X								
Dated_effectivity	X								
Datum	X								
Datum_feature	X								
Datum_reference	X								
Datum_target	X								
Defect_prediction				X			X		
Defining_action_method_relationship	X								
Definitional_representation	X								
Degenerate_toroidal_surface	X								
Derived_shape_aspect	X								
Derived_unit	X								
Derived_unit_element	X								
Description_attribute	X								
Descriptive_representation_item	X								
Design_part	X								
Design_reference		X	X	X	X	X	X	X	X
Die_cast_master			X	X	X	X	X	X	X

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Die_cast_tooling			X	X	X	X	X	X	X
Die_clamping			X	X	X	X	X	X	X
Die_core			X	X	X	X	X	X	X
Die_design_feature			X	X	X	X	X	X	X
Die_master			X	X	X	X	X	X	X
Die_mould_tooling			X	X	X	X	X	X	X
Die_mould_vent			X	X	X	X	X	X	X
Dimension_related_tolerance_zone_element	X								
Dimensional_characteristic_representation	X								
Dimensional_exponents	X								
Dimensional_location	X								
Dimensional_location_with_path	X								
Dimensional_measurement_inspection	X								X
Dimensional_size	X								
Dimensional_size_with_path	X								
Directed_action	X								
Directed_dimensional_location	X								
Direction	X								
Direction_shape_representation (AIC 522)	X								
Document	X								
Document_file	X								
Document_file_properties	X								
Document_product_association		X	X	X	X	X	X	X	X
Document_reference	X								
Document_relationship	X								
Document_representation_type	X								
Document_type	X								
Document_usage_constraint	X								
Document_usage_constraint_assignment	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Document_usage_role	X								
Drafted_surface	X								
Dryer					X			X	X
Edge	X								
Edge_curve	X								
Edge_loop	X								
Edge_round (AIC 522)	X								
Edge_shape_representation	X								
Effectivity									X
Effectivity_assignment									X
Ejector_box			X	X	X	X		X	X
Ejector_design_feature			X	X	X	X		X	X
Ejector_equipment					X			X	X
Ejector_pin			X	X	X	X		X	X
Ejector_plate			X	X	X	X		X	X
Ejector_system			X	X	X	X		X	X
Elementary_surface	X								
Elevated_temperature_service		X	X	X	X		X	X	
Ellipse	X								
Equipment_setting_record								X	
Equipment_usage								X	
Executed_action	X								
Extension	X								
External_identification_assignment	X								
External_source	X								
Externally_defined_class	X								
Externally_defined_dimension_definition	X								
Externally_defined_feature_definition (AIC 522)	X								
Externally_defined_general_property	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Externally_defined_item	X								
Externally_defined_item_relationship	X								
Externally_defined_representation_with_parameters	X								
Externally_defined_schema		X	X	X	X	X	X	X	X
Fabrication									X
Face	X								
Face_bound	X								
Face_outer_bound	X								
Face_shape_representation (AIC 522)	X								
Face_shape_representation_relationship	X								
Face_surface	X								
Faceted_brep	X								
Feature_component_definition (AIC 522)	X								
Feature_component_relationship (AIC 522)	X								
Feature_definition (AIC 522)	X								
Feature_pattern (AIC 522)	X								
Feeder			X	X	X	X		X	X
Fillet (AIC 522)	X								
Filter			X	X	X	X		X	X
Finishing_and_machining_operation		X	X	X	X		X	X	
Finishing_equipment					X			X	X
First_article								X	
Fixture_tab			X	X	X	X		X	X
Flask			X	X	X	X		X	X
Flaskless			X	X	X	X		X	X
Flat_face (AIC 522)	X								
Flatness_tolerance (AIC 519)	X								
Founded_item	X								
Functionally_defined_transformation	X								



**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Gating_design_feature			X	X	X	X		X	X
Gating_system			X	X	X	X		X	X
Gear (AIC 522)	X								
General_property	X								
General_property_association	X								
Generic_manufacturing_resource					X			X	X
Geometric_alignment	X								
Geometric_intersection	X								
Geometric_representation_context	X								
Geometric_representation_item	X								
Geometric_tolerance	X								
Geometric_tolerance_relationship	X								
Geometric_tolerance_with_datum_reference	X								
Geometric_tolerance_with_defined_unit	X								
Global_uncertainty_assigned_context	X								
Global_unit_assigned_context	X								
Group	X								
Group_assignment	X								
Group_relationship	X								
Heat_treat									X
Heat_treat_requirement		X	X	X	X		X	X	
Hole_bottom (AIC 522)	X								
Hyperbola	X								
Id_attribute	X								
Identification_assignment	X								
Identification_role	X								
Illustration		X	X	X	X	X	X	X	X
Impregnation_activity									X
In_facility_location					X			X	X

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Ingate			X	X	X	X		X	X
Ingate_contact			X	X	X	X		X	X
Inspection_activity									X
Inspection_or_test_requirement		X	X	X	X		X	X	
Inspection_or_test_result								X	
Inspection_plan		X	X	X	X		X	X	
Instanced_feature (AIC 522)	X								
Integration_interval				X			X		
Investment_casting_master			X	X	X	X	X	X	X
Investment_casting_tooling			X	X	X	X	X	X	X
Investment_casting_tooling_definition			X	X	X	X	X	X	X
Investment_design_feature			X	X	X	X	X	X	X
Investment_mould			X	X	X	X	X	X	X
Known_source	X								
Length_measure_with_unit	X								
Length_unit	X								
Library_class_version_assignment	X								
Library_property_version_assignment	X								
Limits_and_fits	X								
Line	X								
Line_profile_tolerance (AIC 519)	X								
Linear_profile (AIC 522)	X								
Local_time	X								
Location_shape_representation (AIC 522)	X								
Loop	X								
Loose_piece			X	X	X	X		X	X
Lost_foam_casting_die			X	X	X	X	X	X	X
Lot_effectivity								X	
Machine									X

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Machine_allowance		X	X	X	X		X	X	
Machine_element_relationship									X
Machine_mounting					X			X	X
Machine_setup									X
Machining_activity									X
Machining_process									X
Make_from_usage_option	X								
Manifold_solid_brep	X								
Manufacturing_activity									X
Manufacturing_activity_relationship									X
Manufacturing_process									X
Manufacturing_process_relationship									X
Mapped_item	X								
Marking (AIC 522)	X								
Mass_measure_with_unit	X								
Mass_unit	X								
Master_pattern			X	X	X	X		X	X
Master_sample								X	
Material_designation	X								
Material_property	X								
Material_property_representation	X								
Material_structure								X	
Material_usage								X	
Material_usage_record								X	
Measure_qualification	X								
Measure_representation_item	X								
Measure_with_unit	X								
Mechanical_property_requirements		X	X	X	X		X	X	
Mechanical_stress		X	X	X	X		X	X	

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Melting_equipment					X			X	X
Meshing_condition				X			X		
Metalcaster_simulation									X
Mismatch_tolerance		X	X	X	X		X	X	
Modified_geometric_tolerance	X								
Modified_pattern (AIC 522)	X								
Mould_box			X	X	X	X		X	X
Mould_lock			X	X	X	X		X	X
Moulding_equipment					X			X	X
Name_attribute	X								
Named_unit	X								
Next_assembly_usage_occurrence	X								
Ngon_closed_profile (AIC 522)	X								
Non_casting_activity									X
Non_casting_equipment_process									X
Non_machining_process									X
Non_permanent_moulding_process									X
Object_role	X								
Open_path_profile (AIC 522)	X								
Open_shell	X								
Ordered_part		X							X
Organization	X								
Organization_assignment	X								
Organization_role	X								
Organizational_address	X								
Organizational_project	X								
Oriented_closed_shell	X								
Oriented_edge	X								
Oriented_face	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Oriented_open_shell	X								
Oriented_path	X								
Oriented_surface	X								
Other_inspection								X	
Outer_round (AIC 522)	X								
Outside_profile (AIC 522)	X								
Parabola	X								
Parallel_offset	X								
Parallelism_tolerance (AIC 519)	X								
Parametric_representation_context	X								
Partial_circular_profile (AIC 522)	X								
Parting_surface	X								
Path	X								
Path_feature_component (AIC 522)	X								
Path_shape_representation (AIC 522)	X								
Pattern_equipment_available		X	X	X	X		X	X	
Pattern_offset_membership (AIC 522)	X								
Pattern_omit_membership (AIC 522)	X								
Pattern_plate			X	X	X	X		X	X
Payment_and_shipping		X	X	X	X		X	X	
Pcurve	X								
Permanent_moulding_process									X
Perpendicular_to	X								
Perpendicularity_tolerance (AIC 519)	X								
Person	X								
Person_and_organization	X								
Person_and_organization_assignment	X								
Person_and_organization_role	X								
Personal_address	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Pin_center			X	X	X	X		X	X
Pin_tip			X	X	X	X		X	X
Placed_datum_target_feature	X								
Placement	X								
Planar_shape_representation (AIC 522)	X								
Plane	X								
Plane_angle_measure_with_unit	X								
Plane_angle_unit	X								
Plus_minus_tolerance	X								
Pocket (AIC 522)	X								
Pocket_bottom (AIC 522)	X								
Point	X								
Poly_loop	X								
Polyline	X								
Position_tolerance (AIC 519)	X								
Pre_defined_item	X								
Precision_qualifier	X								
Process_parameter_record								X	
Process_plan_activity			X	X	X	X	X		X
Process_plan_security									X
Process_plan_specification									X
Process_plan_version			X	X	X	X	X		X
Process_product_association									X
Process_property_association									X
Process_requirement		X	X	X	X		X	X	
Product	X								
Product_category	X								
Product_context	X								
Product_definition	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Product_definition_context	X								
Product_definition_effectivity	X								
Product_definition_formation	X								
Product_definition_formation_relationship	X								
Product_definition_process									
Product_definition_relationship	X								
Product_definition_shape	X								
Product_definition_usage	X								
Product_definition_with_associated_documents	X								
Product_material_composition_relationship							X		
Product_related_product_category	X								
Production_core_box			X	X	X	X		X	X
Production_die_cast_mould			X	X	X	X	X	X	X
Production_die_mould			X	X	X	X	X	X	X
Production_investment_cast_mould			X	X	X	X	X	X	X
Production_pattern_definition			X	X	X	X		X	X
Production_rate									X
Production_tool			X	X	X	X	X	X	X
Production_welding									X
Profile_floor (AIC 522)	X								
Projected_zone_definition	X								
Property_definition	X								
Property_definition_relationship	X								
Property_definition_representation	X								
Property_inspection								X	
Property_process	X								
Property_relationship	X								
Protrusion (AIC 522)	X								
Qualified_representation_item	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Quasi_uniform_curve	X								
Quasi_uniform_surface	X								
Ratio_measure_with_unit	X								
Ratio_unit	X								
Rational_b_spline_curve	X								
Rational_b_spline_surface	X								
Rectangular_closed_profile (AIC 522)	X								
Rectangular_pattern (AIC 522)	X								
Red_lined_part		X							
Referenced_modified_datum	X								
Relationship_condition	X								
Removal_volume (AIC 522)	X								
Replicate_feature (AIC 522)	X								
Reporting_requirement		X	X	X	X		X	X	
Representation	X								
Representation_context	X								
Representation_item	X								
Representation_map	X								
Representation_relationship	X								
Request_for_quotation		X							
Requirement_for_action_resource		X	X	X	X	X	X	X	X
Reset_or_push_pin			X	X	X	X		X	X
Resource_property		X	X	X	X	X	X	X	X
Resource_property_representation		X	X	X	X	X	X	X	X
Resource_requirement_type		X	X	X	X	X	X	X	X
Resource_with_material					X			X	X
Resource_with_representation					X			X	X
Revision									X
Revolved_profile (AIC 522)	X								



**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Rib			X	X	X	X		X	X
Rib_top (AIC 522)	X								
Rib_top_floor (AIC 522)	X								
Riser			X	X	X	X		X	X
Riser_contact			X	X	X	X		X	X
Role_association	X								
Round_hole (AIC 522)	X								
Rounded_end (AIC 522)	X								
Rounded_u_profile (AIC 522)	X								
Roundness_tolerance (AIC 519)	X								
Runner			X	X	X	X		X	X
Runout_zone_definition	X								
Runout_zone_orientation	X								
Runout_zone_orientation_reference_direction	X								
Sampled_set								X	
Sand_cast_design_feature			X	X	X	X		X	X
Sand_casting_tooling			X	X	X	X		X	X
Sand_mould			X	X	X	X		X	X
Secondary_tooling			X	X	X	X	X	X	X
Security_classification	X								
Security_classification_assignment	X								
Security_classification_level	X								
Sequential_method									X
Serial_action_method									X
Shakeout					X			X	X
Shape_aspect	X								
Shape_aspect_deriving_relationship	X								
Shape_aspect_relationship	X								
Shape_defining_relationship (AIC 522)	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Shape_definition_representation	X								
Shape_dimension								X	
Shape_dimension_representation	X								
Shape_representation	X								
Shape_representation_with_parameters (AIC 522)	X								
Shot_sleeve			X	X	X	X	X	X	X
Si_unit	X								
Simulated_property				X			X		
Simulation_input				X			X		
Simulation_input_region				X			X		
Simulation_output				X			X		
Simulation_output_region				X			X		
Simulation_process				X			X		
Simulation_result				X			X		
Simulation_run				X			X		
Simulation_run_relationship				X			X		
Simulation_software				X			X		
Simulation_unit				X			X		
Simulation_unit_state				X			X		
Single_activity_relationship				X			X		
Slide			X	X	X	X	X	X	X
Slot (AIC 522)	X								
Slot_end (AIC 522)	X								
Solid_angle_unit	X								
Solid_model	X								
Special_inspection_requirement		X	X	X	X		X	X	
Spherical_cap (AIC 522)	X								
Spherical_surface	X								
Sprue			X	X	X	X		X	X

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Sprue_and_runner_mould			X	X	X	X	X	X	X
Square_u_profile (AIC 522)	X								
Standard_uncertainty	X								
Step (AIC 522)	X								
Stop_pin			X	X	X	X		X	X
Straightness_tolerance (AIC 519)	X								
Surface	X								
Surface_curve	X								
Surface_finish									X
Surface_of_linear_extrusion	X								
Surface_of_revolution	X								
Surface_profile_tolerance (AIC 519)	X								
Swept_surface	X								
Symmetric_shape_aspect	X								
Symmetry_tolerance (AIC 519)	X								
Tangent	X								
Taper (AIC 522)	X								
Tee_profile (AIC 522)	X								
Thread (AIC 522)	X								
Time_assignment	X								
Time_measure_with_unit	X								
Time_role	X								
Time_unit	X								
Tolerance_inspection								X	
Tolerance_requirement		X	X	X	X		X	X	
Tolerance_value	X								
Tolerance_zone	X								
Tolerance_zone_definition	X								
Tolerance_zone_form	X								

**Table 2 — Conformance class elements (continued)**

AIM element	Class								
	1	2	3	4	5	6	7	8	9
Tool_verification								X	
Tooling_process									X
Topological_representation_item	X								
Toroidal_surface	X								
Total_runout_tolerance (AIC 519)	X								
Transition_feature (AIC 522)	X								
Turned_knurl (AIC 522)	X								
Type_qualifier	X								
Uncertainty_measure_with_unit	X								
Uncertainty_qualifier	X								
Uniform_curve	X								
Uniform_surface	X								
Value_range	X								
Value_representation_item	X								
Vector	X								
Vee_profile (AIC 522)	X								
Vent			X	X	X	X		X	X
Versioned_action_request	X								
Vertex	X								
Vertex_loop	X								
Vertex_point	X								
View_reference	X								
Well			X	X	X	X		X	X

## Annex A

### (normative)

### AIM EXPRESS expanded listing

The following EXPRESS is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used. The AIM EXPRESS schema shall have an information object identifier, given in Annex E.

```
(*
cast_parts_schema

*)
SCHEMA cast_parts_schema;
(*
CONSTANT
*)
  CONSTANT
  dummy_gri : geometric_representation_item := representation_item('') ||
             geometric_representation_item();
  dummy_tri : topological_representation_item := representation_item('')
             || topological_representation_item();
  END_CONSTANT;
(*
action_item
*)
  TYPE action_item = SELECT
    (action_resource,
     characterized_object,
     product_definition,
     product_definition_formation,
     property_definition);
  END_TYPE; -- action_item
(*
action_method_item
*)
  TYPE action_method_item = SELECT
    (shape_aspect,
     externally_defined_item,
     product_definition,
     property_definition);
  END_TYPE; -- action_method_item
(*
action_request_item
*)
  TYPE action_request_item = SELECT
```

```
(document,
  product_definition,
  property_definition);
END_TYPE; -- action_request_item
(*)
ahead_or_behind
*)
  TYPE ahead_or_behind = ENUMERATION OF
    (exact,
     ahead,
     behind);
  END_TYPE; -- ahead_or_behind
(*)
amount_of_substance_measure
*)
  TYPE amount_of_substance_measure = REAL;
  END_TYPE; -- amount_of_substance_measure
(*)
angle_relator
*)
  TYPE angle_relator = ENUMERATION OF
    (equal,
     small,
     large);
  END_TYPE; -- angle_relator
(*)
approved_item
*)
  TYPE approved_item = SELECT
    (action_method,
     action_relationship,
     directed_action,
     product_definition_formation,
     versioned_action_request);
  END_TYPE; -- approved_item
(*)
area_measure
*)
  TYPE area_measure = REAL;
  END_TYPE; -- area_measure
(*)
attribute_type
*)
  TYPE attribute_type = SELECT
    (label,
     text);
  END_TYPE; -- attribute_type
(*)
axis2_placement
```

```

*)
  TYPE axis2_placement = SELECT
    (axis2_placement_2d,
     axis2_placement_3d);
  END_TYPE; -- axis2_placement
(*)
b_spline_curve_form
*)
  TYPE b_spline_curve_form = ENUMERATION OF
    (elliptic_arc,
     polyline_form,
     parabolic_arc,
     circular_arc,
     unspecified,
     hyperbolic_arc);
  END_TYPE; -- b_spline_curve_form
(*)
b_spline_surface_form
*)
  TYPE b_spline_surface_form = ENUMERATION OF
    (surf_of_linear_extrusion,
     plane_surf,
     generalised_cone,
     toroidal_surf,
     conical_surf,
     spherical_surf,
     unspecified,
     ruled_surf,
     surf_of_revolution,
     cylindrical_surf,
     quadric_surf);
  END_TYPE; -- b_spline_surface_form
(*)
boolean_operand
*)
  TYPE boolean_operand = SELECT
    (solid_model);
  END_TYPE; -- boolean_operand
(*)
celsius_temperature_measure
*)
  TYPE celsius_temperature_measure = REAL;
  END_TYPE; -- celsius_temperature_measure
(*)
characterized_action_definition
*)
  TYPE characterized_action_definition = SELECT
    (action,
     action_method,
     action_method_relationship,
     action_relationship);

```

```
END_TYPE; -- characterized_action_definition
(*)
```

### **characterized\_definition**

```
*)
TYPE characterized_definition = SELECT
  (characterized_object,
   characterized_product_definition,
   shape_definition);
END_TYPE; -- characterized_definition
(*)
```

### **characterized\_material\_property**

```
*)
TYPE characterized_material_property = SELECT
  (material_property_representation);
END_TYPE; -- characterized_material_property
(*)
```

### **characterized\_product\_definition**

```
*)
TYPE characterized_product_definition = SELECT
  (product_definition,
   product_definition_relationship);
END_TYPE; -- characterized_product_definition
(*)
```

### **characterized\_resource\_definition**

```
*)
TYPE characterized_resource_definition = SELECT
  (action_resource,
   action_resource_relationship,
   action_resource_requirement);
END_TYPE; -- characterized_resource_definition
(*)
```

### **classification\_item**

```
*)
TYPE classification_item = SELECT
  (externally_defined_representation_with_parameters);
END_TYPE; -- classification_item
(*)
```

### **compound\_item\_definition**

```
*)
TYPE compound_item_definition = SELECT
  (set_representation_item);
END_TYPE; -- compound_item_definition
(*)
```

### **count\_measure**

```
*)
TYPE count_measure = NUMBER;
END_TYPE; -- count_measure
(*)
```

### **curve\_on\_surface**

```
*)
```



```

TYPE curve_on_surface = SELECT
    (pcurve,
     surface_curve,
     composite_curve_on_surface);
END_TYPE; -- curve_on_surface
(*)
date_and_time_item
*)
TYPE date_and_time_item = SELECT
    (representation);
END_TYPE; -- date_and_time_item
(*)
date_item
*)
TYPE date_item = SELECT
    (action,
     action_resource,
     product_definition,
     property_definition,
     process_plan_security,
     document,
     versioned_action_request);
END_TYPE; -- date_item
(*)
date_time_or_event_occurrence
*)
TYPE date_time_or_event_occurrence = SELECT
    (date_time_select);
END_TYPE; -- date_time_or_event_occurrence
(*)
date_time_select
*)
TYPE date_time_select = SELECT
    (date,
     local_time,
     date_and_time);
END_TYPE; -- date_time_select
(*)
day_in_month_number
*)
TYPE day_in_month_number = INTEGER;
WHERE
    wr1: (1 <= SELF) AND (SELF <= 31);
END_TYPE; -- day_in_month_number
(*)
derived_property_select
*)
TYPE derived_property_select = SELECT
    (property_definition,
     action_property,
     resource_property);

```

```
END_TYPE; -- derived_property_select
(*)
description_attribute_select
*)
TYPE description_attribute_select = SELECT
  (action_request_solution,
   application_context,
   approval_role,
   date_role,
   date_time_role,
   effectivity,
   external_source,
   organization_role,
   person_and_organization_role,
   person_and_organization,
   property_definition_representation,
   representation,
   time_role);
END_TYPE; -- description_attribute_select
(*)
```

### **dimension\_count**

```
*)
TYPE dimension_count = INTEGER;
WHERE
  wr1: SELF > 0;
END_TYPE; -- dimension_count
(*)
```

### **dimensional\_characteristic**

```
*)
TYPE dimensional_characteristic = SELECT
  (dimensional_location,
   dimensional_size);
END_TYPE; -- dimensional_characteristic
(*)
```

### **document\_reference\_item**

```
*)
TYPE document_reference_item = SELECT
  (action,
   action_method,
   action_resource,
   action_request_solution,
   externally_defined_feature_definition,
   externally_defined_schema,
   property_definition,
   resource_property,
   requirement_for_action_resource,
   known_source);
END_TYPE; -- document_reference_item
(*)
```

### **effectivity\_item**

```

*)
  TYPE effectivity_item = SELECT
    (product_definition,
     property_definition);
  END_TYPE; -- effectivity_item
(*)
electric_current_measure
*)
  TYPE electric_current_measure = REAL;
  END_TYPE; -- electric_current_measure
(*)
external_identification_item
*)
  TYPE external_identification_item = SELECT
    (document,
     externally_defined_class,
     externally_defined_general_property,
     product_definition);
  END_TYPE; -- external_identification_item
(*)
founded_item_select
*)
  TYPE founded_item_select = SELECT
    (founded_item,
     representation_item);
  END_TYPE; -- founded_item_select
(*)
geometric_set_select
*)
  TYPE geometric_set_select = SELECT
    (point,
     curve,
     surface);
  END_TYPE; -- geometric_set_select
(*)
group_item
*)
  TYPE group_item = SELECT
    (instanced_feature,
     replicate_feature,
     transition_feature);
  END_TYPE; -- group_item
(*)
hour_in_day
*)
  TYPE hour_in_day = INTEGER;
  WHERE
    wr1: (0 <= SELF) AND (SELF < 24);
  END_TYPE; -- hour_in_day
(*)

```

### **id\_attribute\_select**

```
*)  
  TYPE id_attribute_select = SELECT  
    (action,  
     address,  
     product_category,  
     property_definition,  
     shape_aspect,  
     shape_aspect_relationship,  
     application_context,  
     group,  
     organizational_project,  
     representation);  
  END_TYPE; -- id_attribute_select  
(*
```

### **identification\_assignment\_item**

```
*)  
  TYPE identification_assignment_item = SELECT  
    (document,  
     document_file);  
  END_TYPE; -- identification_assignment_item  
(*
```

### **identifier**

```
*)  
  TYPE identifier = STRING;  
  END_TYPE; -- identifier  
(*
```

### **knot\_type**

```
*)  
  TYPE knot_type = ENUMERATION OF  
    (uniform_knots,  
     quasi_uniform_knots,  
     piecewise_bezier_knots,  
     unspecified);  
  END_TYPE; -- knot_type  
(*
```

### **label**

```
*)  
  TYPE label = STRING;  
  END_TYPE; -- label  
(*
```

### **length\_measure**

```
*)  
  TYPE length_measure = REAL;  
  END_TYPE; -- length_measure  
(*
```

### **limit\_condition**

```
*)  
  TYPE limit_condition = ENUMERATION OF  
    (regardless_of_feature_size,
```

```

        maximum_material_condition,
        least_material_condition);
    END_TYPE; -- limit_condition
(*)
list_of_reversible_topology_item
*)
    TYPE list_of_reversible_topology_item = LIST [0:?] OF
        reversible_topology_item;
    END_TYPE; -- list_of_reversible_topology_item

```

### **luminous\_intensity\_measure**

```

(*)
    TYPE luminous_intensity_measure = REAL;
    END_TYPE; -- luminous_intensity_measure
(*)

```

### **mass\_measure**

```

(*)
    TYPE mass_measure = REAL;
    END_TYPE; -- mass_measure
(*)

```

### **measure\_value**

```

(*)
    TYPE measure_value = SELECT
        (length_measure,
         mass_measure,
         time_measure,
         electric_current_measure,
         thermodynamic_temperature_measure,
         celsius_temperature_measure,
         amount_of_substance_measure,
         luminous_intensity_measure,
         plane_angle_measure,
         solid_angle_measure,
         area_measure,
         volume_measure,
         ratio_measure,
         parameter_value,
         positive_length_measure,
         positive_plane_angle_measure,
         count_measure);
    END_TYPE; -- measure_value
(*)

```

### **minute\_in\_hour**

```

(*)
    TYPE minute_in_hour = INTEGER;
    WHERE
        wr1: (0 <= SELF) AND (SELF <= 59);
    END_TYPE; -- minute_in_hour
(*)

```

### **month\_in\_year\_number**

```

(*)

```

## ISO 10303-223:2008 (E)

```
TYPE month_in_year_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 12);
END_TYPE; -- month_in_year_number
```

(\*

### **name\_attribute\_select**

\*)

```
TYPE name_attribute_select = SELECT
  (action_request_solution,
   address,
   derived_unit,
   effectivity,
   person_and_organization,
   product_definition,
   property_definition_representation);
END_TYPE; -- name_attribute_select
```

(\*

### **ordered\_item**

\*)

```
TYPE ordered_item = SELECT
  (action,product_definition_formation);
END_TYPE; -- ordered_item
```

(\*

### **organization\_item**

\*)

```
TYPE organization_item = SELECT
  (action_method,
   action_resource,
   known_source,
   property_definition);
END_TYPE; -- organization_item
```

(\*

### **parameter\_value**

\*)

```
TYPE parameter_value = REAL;
END_TYPE; -- parameter_value
```

(\*

### **pcurve\_or\_surface**

\*)

```
TYPE pcurve_or_surface = SELECT
  (pcurve,
   surface);
END_TYPE; -- pcurve_or_surface
```

(\*

### **person\_and\_organization\_item**

\*)

```
TYPE person_and_organization_item = SELECT
  (action,
   product_definition_formation,
   product_definition,
   property_definition,
```

```

        shape_aspect,
        versioned_action_request);
    END_TYPE; -- person_and_organization_item
(*)
person_organization_select
*)
    TYPE person_organization_select = SELECT
        (person,
         organization,
         person_and_organization);
    END_TYPE; -- person_organization_select
(*)
plane_angle_measure
*)
    TYPE plane_angle_measure = REAL;
    END_TYPE; -- plane_angle_measure
(*)
positive_length_measure
*)
    TYPE positive_length_measure = length_measure;
    WHERE
        wr1: SELF > 0;
    END_TYPE; -- positive_length_measure
(*)
positive_plane_angle_measure
*)
    TYPE positive_plane_angle_measure = plane_angle_measure;
    WHERE
        wr1: SELF > 0;
    END_TYPE; -- positive_plane_angle_measure
(*)
preferred_surface_curve_representation
*)
    TYPE preferred_surface_curve_representation = ENUMERATION OF
        (pcurve_s2,
         pcurve_s1,
         curve_3d);
    END_TYPE; -- preferred_surface_curve_representation
(*)
process_or_process_relationship
*)
    TYPE process_or_process_relationship = SELECT
        (product_definition_process,
         property_process,
         relationship_with_condition);
    END_TYPE; -- process_or_process_relationship
(*)
product_or_formation_or_definition
*)
    TYPE product_or_formation_or_definition = SELECT

```

```
        (product,  
         product_definition_formation,  
         product_definition);  
    END_TYPE; -- product_or_formation_or_definition  
(*
```

### **property\_or\_shape\_select**

```
*)  
    TYPE property_or_shape_select = SELECT  
        (property_definition,  
         shape_definition);  
    END_TYPE; -- property_or_shape_select  
(*
```

### **ratio\_measure**

```
*)  
    TYPE ratio_measure = REAL;  
    END_TYPE; -- ratio_measure  
(*
```

### **relationship\_with\_condition**

```
*)  
    TYPE relationship_with_condition = SELECT  
        (action_method_relationship,  
         action_relationship);  
    END_TYPE; -- relationship_with_condition  
(*
```

### **represented\_definition**

```
*)  
    TYPE represented_definition = SELECT  
        (general_property,  
         property_definition,  
         property_definition_relationship,  
         shape_aspect,  
         shape_aspect_relationship);  
    END_TYPE; -- represented_definition  
(*
```

### **reversible\_topology**

```
*)  
    TYPE reversible_topology = SELECT  
        (reversible_topology_item,  
         list_of_reversible_topology_item,  
         set_of_reversible_topology_item);  
    END_TYPE; -- reversible_topology  
(*
```

### **reversible\_topology\_item**

```
*)  
    TYPE reversible_topology_item = SELECT  
        (edge,  
         path,  
         face,  
         face_bound,  
         closed_shell,  
         open_shell);
```



```

    END_TYPE; -- reversible_topology_item
  (*)
role_select
  *)
    TYPE role_select = SELECT
      (action_assignment,
       action_request_assignment,
       approval_assignment,
       approval_date_time,
       document_reference,
       effectivity_assignment,
       group_assignment,
       security_classification_assignment);
    END_TYPE; -- role_select
  (*)
second_in_minute
  *)
    TYPE second_in_minute = REAL;
    WHERE
      wr1: (0 <= SELF) AND (SELF <= 60);
    END_TYPE; -- second_in_minute
  (*)
security_classification_item
  *)
    TYPE security_classification_item = SELECT
      (action,
       action_method,
       product_definition_formation);
    END_TYPE; -- security_classification_item
  (*)
set_of_reversible_topology_item
  *)
    TYPE set_of_reversible_topology_item = SET [0:?] OF
      reversible_topology_item;
    END_TYPE; -- set_of_reversible_topology_item
  (*)
set_representation_item
  *)
    TYPE set_representation_item = SET [1:?] OF representation_item;
    END_TYPE; -- set_representation_item
  (*)
shape_definition
  *)
    TYPE shape_definition = SELECT
      (product_definition_shape,
       shape_aspect,
       shape_aspect_relationship);
    END_TYPE; -- shape_definition
  (*)
shell

```

```
*)  
  TYPE shell = SELECT  
    (open_shell,  
     closed_shell);  
  END_TYPE; -- shell
```

(\*

### **si\_prefix**

\*)

```
  TYPE si_prefix = ENUMERATION OF  
    (exa,  
     pico,  
     mega,  
     femto,  
     atto,  
     centi,  
     nano,  
     hecto,  
     micro,  
     tera,  
     giga,  
     milli,  
     peta,  
     deci,  
     kilo,  
     deca);  
  END_TYPE; -- si_prefix
```

(\*

### **si\_unit\_name**

\*)

```
  TYPE si_unit_name = ENUMERATION OF  
    (hertz,  
     degree_celsius,  
     siemens,  
     sievert,  
     lux,  
     watt,  
     ohm,  
     second,  
     becquerel,  
     pascal,  
     henry,  
     tesla,  
     volt,  
     joule,  
     kelvin,  
     ampere,  
     gram,  
     steradian,  
     mole,  
     lumen,  
     gray,  
     candela,  
     farad,
```

```

        radian,
        newton,
        metre,
        weber,
        coulomb);
    END_TYPE; -- si_unit_name
(*)
solid_angle_measure
*)
    TYPE solid_angle_measure = REAL;
    END_TYPE; -- solid_angle_measure
(*)
source_item
*)
    TYPE source_item = SELECT
        (identifier);
    END_TYPE; -- source_item
(*)
supported_item
*)
    TYPE supported_item = SELECT
        (action_directive,
         action,
         action_method);
    END_TYPE; -- supported_item
(*)
text
*)
    TYPE text = STRING;
    END_TYPE; -- text
(*)
thermodynamic_temperature_measure
*)
    TYPE thermodynamic_temperature_measure = REAL;
    END_TYPE; -- thermodynamic_temperature_measure
(*)
time_item
*)
    TYPE time_item = SELECT
        (action_property,
         action_request_solution);
    END_TYPE; -- time_item
(*)
time_measure
*)
    TYPE time_measure = REAL;
    END_TYPE; -- time_measure
(*)
tolerance_method_definition
*)

```

```

    TYPE tolerance_method_definition = SELECT
        (tolerance_value,
         limits_and_fits);
    END_TYPE; -- tolerance_method_definition
(*)
tolerance_select
*)
    TYPE tolerance_select = SELECT
        (geometric_tolerance,
         plus_minus_tolerance);
    END_TYPE; -- tolerance_select
(*)
transformation
*)
    TYPE transformation = SELECT
        (functionally_defined_transformation);
    END_TYPE; -- transformation
(*)
transition_code
*)
    TYPE transition_code = ENUMERATION OF
        (discontinuous,
         cont_same_gradient_same_curvature,
         cont_same_gradient,
         continuous);
    END_TYPE; -- transition_code
(*)
trimming_select
*)
    TYPE trimming_select = SELECT
        (cartesian_point,
         parameter_value);
    END_TYPE; -- trimming_select
(*)
unit
*)
    TYPE unit = SELECT
        (named_unit,
         derived_unit);
    END_TYPE; -- unit
(*)
value_qualifier
*)
    TYPE value_qualifier = SELECT
        (precision_qualifier,
         type_qualifier,
         uncertainty_qualifier);
    END_TYPE; -- value_qualifier
(*)
vector_or_direction

```

```

*)
  TYPE vector_or_direction = SELECT
    (vector,
     direction);
  END_TYPE; -- vector_or_direction
(*)
volume_measure
*)
  TYPE volume_measure = REAL;
  END_TYPE; -- volume_measure
(*)
year_number
*)
  TYPE year_number = INTEGER;
  END_TYPE; -- year_number
(*)
action
*)
  ENTITY action;
    name          : label;
    description   : OPTIONAL text;
    chosen_method  : action_method;
  DERIVE
    id : identifier := get_id_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
                      'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  END_ENTITY; -- action
(*)
action_assignment
*)
  ENTITY action_assignment
  ABSTRACT SUPERTYPE;
    assigned_action : action;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
                      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
  END_ENTITY; -- action_assignment
(*)
action_directive
*)
  ENTITY action_directive;
    name          : label;
    description   : OPTIONAL text;
    analysis      : text;
    comment       : text;
    requests      : SET [1:?] OF versioned_action_request;
  END_ENTITY; -- action_directive
(*)

```

### **action\_method**

```
*)  
  ENTITY action_method;  
    name          : label;  
    description   : OPTIONAL text;  
    consequence  : text;  
    purpose      : text;  
  END_ENTITY; -- action_method
```

(\*

### **action\_method\_assignment**

```
*)  
  ENTITY action_method_assignment  
    ABSTRACT SUPERTYPE;  
    assigned_action_method : action_method;  
    role                   : action_method_role;  
  END_ENTITY; -- action_method_assignment
```

(\*

### **action\_method\_relationship**

```
*)  
  ENTITY action_method_relationship;  
    name          : label;  
    description   : OPTIONAL text;  
    relating_method : action_method;  
    related_method : action_method;  
  END_ENTITY; -- action_method_relationship
```

(\*

### **action\_method\_role**

```
*)  
  ENTITY action_method_role;  
    name          : label;  
    description   : OPTIONAL text;  
  END_ENTITY; -- action_method_role
```

(\*

### **action\_method\_with\_associated\_documents**

```
*)  
  ENTITY action_method_with_associated_documents  
    SUBTYPE OF (action_method);  
    documents : SET [1:?] OF document;  
  END_ENTITY; -- action_method_with_associated_documents
```

(\*

### **action\_property**

```
*)  
  ENTITY action_property;  
    name          : label;  
    description   : text;  
    definition    : characterized_action_definition;  
  END_ENTITY; -- action_property
```

(\*

### **action\_property\_relationship**

\*)

```

ENTITY action_property_relationship;
    name                : label;
    description         : text;
    relating_action_property : action_property;
    related_action_property  : action_property;
    WHERE
        wr1: relating_action_property :<>: related_action_property;
END_ENTITY; -- action_property_relationship

```

(\*

### **action\_property\_representation**

\*)

```

ENTITY action_property_representation;
    name                : label;
    description         : text;
    property            : action_property;
    representation     : representation;
END_ENTITY; -- action_property_representation

```

(\*

### **action\_relationship**

\*)

```

ENTITY action_relationship;
    name                : label;
    description         : OPTIONAL text;
    relating_action     : action;
    related_action      : action;
END_ENTITY; -- action_relationship

```

(\*

### **action\_request\_assignment**

\*)

```

ENTITY action_request_assignment
    ABSTRACT SUPERTYPE;
    assigned_action_request : versioned_action_request;
    DERIVE
        role : object_role := get_role(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_request_assignment

```

(\*

### **action\_request\_solution**

\*)

```

ENTITY action_request_solution;
    method : action_method;
    request : versioned_action_request;
    DERIVE
        description : text := get_description_value(SELF);
        name        : label := get_name_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
        wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +

```

```

        'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
    END_ENTITY; -- action_request_solution
(*)
action_resource
*)
    ENTITY action_resource;
        name          : label;
        description    : OPTIONAL text;
        usage          : SET [1:?] OF supported_item;
        kind           : action_resource_type;
    END_ENTITY; -- action_resource
(*)
action_resource_relationship
*)
    ENTITY action_resource_relationship;
        name          : label;
        description    : OPTIONAL text;
        relating_resource : action_resource;
        related_resource  : action_resource;
    END_ENTITY; -- action_resource_relationship
(*)
action_resource_requirement
*)
    ENTITY action_resource_requirement;
        name          : label;
        description    : text;
        kind           : resource_requirement_type;
        OPERATIONS    : SET [1:?] OF characterized_action_definition;
    END_ENTITY; -- action_resource_requirement
(*)
action_resource_type
*)
    ENTITY action_resource_type;
        name : label;
    END_ENTITY; -- action_resource_type
(*)
action_status
*)
    ENTITY action_status;
        status          : label;
        assigned_action : executed_action;
    END_ENTITY; -- action_status
(*)
activity_execution_result
*)
    ENTITY activity_execution_result
        SUBTYPE OF (executed_action);
        WHERE
            wr1: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') | (SIZEOF(
                QUERY ( apr <* USEDIN(ap,

```



```

        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | ('CAST_PARTS_SCHEMA.ACTIVITY_RESULT_RECORD' IN TYPEOF(apr
        .representation)) )) = 1) )) = 0);
wr2: SIZEOF(QUERY ( ppa <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
        ((ppa.description = 'validation part shape') AND (
        'CAST_PARTS_SCHEMA.SAMPLED_SET' IN TYPEOF(ppa.
        property_or_shape))) )) = 1;
END_ENTITY; -- activity_execution_result
(*)
activity_result_record
*)
ENTITY activity_result_record
  SUBTYPE OF (representation);
  WHERE
    wr1: SIZEOF(QUERY ( dt <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_DATE_AND_TIME_ASSIGNMENT.ITEMS')
        | (dt\date_and_time_assignment.role.name =
        'production date') )) = 1;
END_ENTITY; -- activity_result_record
(*)
address
*)
ENTITY address;

    internal_location      : OPTIONAL label;
    street_number          : OPTIONAL label;
    street                  : OPTIONAL label;
    postal_box              : OPTIONAL label;
    town                    : OPTIONAL label;
    region                  : OPTIONAL label;
    postal_code             : OPTIONAL label;
    country                 : OPTIONAL label;
    facsimile_number        : OPTIONAL label;
    telephone_number        : OPTIONAL label;
    electronic_mail_address : OPTIONAL label;
    telex_number            : OPTIONAL label;
  DERIVE
    name : label := get_name_value(SELF);
    url  : identifier := get_id_value(SELF);
  WHERE
    wr1: ((((((((((EXISTS(internal_location) OR EXISTS(street_number))
    OR EXISTS(street)) OR EXISTS(postal_box)) OR EXISTS(town))
    OR EXISTS(region)) OR EXISTS(postal_code)) OR EXISTS(country))
    OR EXISTS(facsimile_number)) OR EXISTS(telephone_number)) OR
    EXISTS(electronic_mail_address)) OR EXISTS(telex_number);
END_ENTITY; -- address
(*)
advanced_brep_shape_representation
*)
ENTITY advanced_brep_shape_representation

```

```

SUBTYPE OF (shape_representation);
WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
    'CAST_PARTS_SCHEMA.MANIFOLD_SOLID_BREP',
    'CAST_PARTS_SCHEMA.FACETED_BREP',
    'CAST_PARTS_SCHEMA.MAPPED_ITEM',
    'CAST_PARTS_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it)) = 1)) ))
    = 0;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
    'CAST_PARTS_SCHEMA.MANIFOLD_SOLID_BREP',
    'CAST_PARTS_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1)) )) > 0;
  wr3: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
    'CAST_PARTS_SCHEMA.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
    NOT (SIZEOF(QUERY ( csh <* msb_shells(msb) | (NOT (SIZEOF(
    QUERY ( fcs <* csh\connected_face_set.cfs_faces | (NOT (
    'CAST_PARTS_SCHEMA.ADVANCED_FACE' IN TYPEOF(fcs))) )) = 0)) ))
    = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
    'CAST_PARTS_SCHEMA.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) ) | (
    'CAST_PARTS_SCHEMA.ORIENTED_CLOSED_SHELL' IN TYPEOF(msb\
    manifold_solid_brep.outer)) )) = 0;
  wr5: SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
    'CAST_PARTS_SCHEMA.BREP_WITH_VOIDS' IN TYPEOF(it)) ) | (NOT
    (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids | csh\
    oriented_closed_shell.orientation )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
    'CAST_PARTS_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (NOT (
    'CAST_PARTS_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION' IN
    TYPEOF(mi\mapped_item.mapping_source.mapped_representation)))
  ))
    = 0;
END_ENTITY; -- advanced_brep_shape_representation

```

## advanced\_face

```

*)
ENTITY advanced_face
SUBTYPE OF (face_surface);
WHERE
  wr1 : SIZEOF(['CAST_PARTS_SCHEMA.ELEMENTARY_SURFACE',
    'CAST_PARTS_SCHEMA.B_SPLINE_SURFACE',
    'CAST_PARTS_SCHEMA.SWEPT_SURFACE'] * TYPEOF(face_geometry))
    = 1;
  wr2 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
    'CAST_PARTS_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
    NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list |
    (NOT ('CAST_PARTS_SCHEMA.EDGE_CURVE' IN TYPEOF(oe\
    oriented_edge.edge_element))) )) = 0)) )) = 0;
  wr3 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
    'CAST_PARTS_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
    NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list |
    (NOT (SIZEOF(['CAST_PARTS_SCHEMA.LINE',
    'CAST_PARTS_SCHEMA.CONIC', 'CAST_PARTS_SCHEMA.POLYLINE',
    'CAST_PARTS_SCHEMA.SURFACE_CURVE',

```

```

'CAST_PARTS_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(oe.
edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
0;
wr4 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
'CAST_PARTS_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list |
(NOT (((('CAST_PARTS_SCHEMA.VERTEX_POINT' IN TYPEOF(oe\edge.
edge_start)) AND ('CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN
TYPEOF(oe\edge.edge_start\vertex_point.vertex_geometry)))
AND (('CAST_PARTS_SCHEMA.VERTEX_POINT' IN TYPEOF(oe\edge.
edge_end)) AND ('CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN
TYPEOF(oe\edge.edge_end\vertex_point.vertex_geometry)))))) ))
= 0)) )) = 0;
wr5 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
'CAST_PARTS_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
'CAST_PARTS_SCHEMA.ORIENTED_PATH' IN TYPEOF(elp_fbnds.bound)) ))
= 0;
wr6 : (NOT ('CAST_PARTS_SCHEMA.SWEPT_SURFACE' IN TYPEOF(
face_geometry))) OR (SIZEOF(['CAST_PARTS_SCHEMA.LINE',
'CAST_PARTS_SCHEMA.CONIC', 'CAST_PARTS_SCHEMA.POLYLINE',
'CAST_PARTS_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(face_geometry\
swept_surface.swept_curve)) = 1);
wr7 : SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* bounds | (
'CAST_PARTS_SCHEMA.VERTEX_LOOP' IN TYPEOF(bnds.bound)) ) |
(NOT (('CAST_PARTS_SCHEMA.VERTEX_POINT' IN TYPEOF(vlp_fbnds
\face_bound.bound\vertex_loop.loop_vertex)) AND (
'CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN TYPEOF(vlp_fbnds\
face_bound.bound\vertex_loop.loop_vertex\vertex_point.
vertex_geometry)))))) )) = 0;
wr8 : SIZEOF(QUERY ( bnd <* bounds | (NOT (SIZEOF([
'CAST_PARTS_SCHEMA.EDGE_LOOP',
'CAST_PARTS_SCHEMA.VERTEX_LOOP'] * TYPEOF(bnd.bound)) = 1)) ))
= 0;
wr9 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
'CAST_PARTS_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) ) | (
NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.edge_list |
(('CAST_PARTS_SCHEMA.SURFACE_CURVE' IN TYPEOF(oe\
oriented_edge.edge_element\edge_curve.edge_geometry)) AND (
NOT (SIZEOF(QUERY ( sc_ag <* oe.edge_element\edge_curve.
edge_geometry\surface_curve.associated_geometry | (NOT (
'CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(sc_ag)) )) = 0)) )) ))
= 0)) )) = 0;
wr10: ((NOT ('CAST_PARTS_SCHEMA.SWEPT_SURFACE' IN TYPEOF(
face_geometry))) OR ((NOT ('CAST_PARTS_SCHEMA.POLYLINE' IN
TYPEOF(face_geometry\swept_surface.swept_curve))) OR (
SIZEOF(face_geometry\swept_surface.swept_curve\polyline.
points) >= 3))) AND (SIZEOF(QUERY ( elp_fbnds <*
QUERY ( bnds <* bounds | ('CAST_PARTS_SCHEMA.EDGE_LOOP' IN
TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
elp_fbnds.bound\path.edge_list | ((
'CAST_PARTS_SCHEMA.POLYLINE' IN TYPEOF(oe\oriented_edge.
edge_element\edge_curve.edge_geometry)) AND (NOT (SIZEOF(oe
\oriented_edge.edge_element\edge_curve.edge_geometry\

```

```

        polyline.points) >= 3))) )) = 0)) )) = 0);
    END_ENTITY; -- advanced_face
  (*)
allowed_time
  *)
  ENTITY allowed_time
    SUBTYPE OF (action_property);
    WHERE
      wr1: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (SIZEOF(QUERY ( it <* apr.representation.items | ((
        SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(it)) =
        2) AND (it.name = 'allowance factor')) )) = 1) )) = 0);
      wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (SIZEOF(QUERY ( it <* apr.representation.items | ((
        SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(it)) =
        2) AND (it.name = 'standard time')) )) = 1) )) = 0);
      wr3: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (SIZEOF(QUERY ( it <* apr.representation.items | ((
        'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'allowed type')) )) <= 1) )) = 0);
      wr4: SIZEOF(QUERY ( apr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (SIZEOF(QUERY ( it <* apr.representation.items | ((
        'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'allowed time source')) )) = 1) ))
        = 1;
    END_ENTITY; -- allowed_time
  (*)

```

### **alternate\_action\_method\_relationship**

```

  *)
  ENTITY alternate_action_method_relationship
    SUBTYPE OF (action_method_relationship);
    WHERE
      wr1: 'CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
        related_method);
      wr2: 'CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
        relating_method);
    END_ENTITY; -- alternate_action_method_relationship
  (*)

```

### **alternate\_plan\_relationship**

```

  *)
  ENTITY alternate_plan_relationship
    SUBTYPE OF (action_relationship);
    WHERE
      wr1: 'CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(SELF.
        related_action);

```

```

        wr2: 'CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(SELF.
            relating_action);
    END_ENTITY; -- alternate_plan_relationship
(*)
ancillary_activity
*)
    ENTITY ancillary_activity
        SUBTYPE OF (process_plan_activity);
    END_ENTITY; -- ancillary_activity
(*)
ancillary_setup
*)
    ENTITY ancillary_setup
        SUBTYPE OF (process_plan_activity);
    END_ENTITY; -- ancillary_setup
(*)
angular_location
*)
    ENTITY angular_location
        SUBTYPE OF (dimensional_location);
        angle_selection : angle_relator;
    END_ENTITY; -- angular_location
(*)
angular_size
*)
    ENTITY angular_size
        SUBTYPE OF (dimensional_size);
        angle_selection : angle_relator;
    END_ENTITY; -- angular_size
(*)
angularity_tolerance
*)
    ENTITY angularity_tolerance
        SUBTYPE OF (geometric_tolerance_with_datum_reference);
        WHERE
            wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
                datum_system) < 3;
    END_ENTITY; -- angularity_tolerance
(*)
apex
*)
    ENTITY apex
        SUBTYPE OF (derived_shape_aspect);
    END_ENTITY; -- apex
(*)
application_context
*)
    ENTITY application_context;

        application : label;

```

```

DERIVE
  description : text := get_description_value(SELF);
  id          : identifier := get_id_value(SELF);
INVERSE
  context_elements : SET [1:?] OF application_context_element FOR
                        frame_of_reference;
WHERE
  wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
                    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
                    'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- application_context
(*)
application_context_element
*)
ENTITY application_context_element
  SUPERTYPE OF (ONEOF (product_context, product_definition_context));
  name          : label;
  frame_of_reference : application_context;
END_ENTITY; -- application_context_element
(*)
application_protocol_definition
*)
ENTITY application_protocol_definition;
  status          : label;
  application_interpreted_model_schema_name : label;
  application_protocol_year : year_number;
  application      : application_context;
END_ENTITY; -- application_protocol_definition
(*)
applied_action_assignment
*)
ENTITY applied_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_item;
END_ENTITY; -- applied_action_assignment
(*)
applied_action_method_assignment
*)
ENTITY applied_action_method_assignment
  SUBTYPE OF (action_method_assignment);
  items : SET [1:?] OF action_method_item;
END_ENTITY; -- applied_action_method_assignment
(*)
applied_action_request_assignment
*)
ENTITY applied_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF action_request_item;
END_ENTITY; -- applied_action_request_assignment
(*)

```

**applied\_approval\_assignment**

```

*)
ENTITY applied_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approved_item;
END_ENTITY; -- applied_approval_assignment

```

```

(*)

```

**applied\_area**

```

*)
ENTITY applied_area
  SUBTYPE OF (shape_aspect);
WHERE
  wr1: 'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(SELF.
    of_shape);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT ((2 <= SIZEOF(impl_rep.
    used_representation.items)) AND (SIZEOF(impl_rep.
    used_representation.items) <= 3))) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
    used_representation.items | (NOT (srwp_i.name IN [
    'orientation','effective length','maximum length']))) )) > 0)) ))
    = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'effective length')) )) = 1)) )) = 0)) ))

```

```

        <= 1;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'maximum length')) ) <= 1)) ) = 0)) )
    = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((
    'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
    'orientation')) ) = 1)) ) = 0)) ) = 0;
END_ENTITY; -- applied_area
(*)
applied_classification_assignment
*)
ENTITY applied_classification_assignment
    SUBTYPE OF (classification_assignment);
    items : SET [1:?] OF classification_item;
END_ENTITY; -- applied_classification_assignment
(*)
applied_date_and_time_assignment
*)
ENTITY applied_date_and_time_assignment
    SUBTYPE OF (date_and_time_assignment);
    items : SET [1:?] OF date_and_time_item;
END_ENTITY; -- applied_date_and_time_assignment
(*)
applied_date_assignment
*)
ENTITY applied_date_assignment
    SUBTYPE OF (date_assignment);
    items : SET [1:?] OF date_item;
END_ENTITY; -- applied_date_assignment
(*)
applied_document_reference
*)
ENTITY applied_document_reference
    SUBTYPE OF (document_reference);

```



```

        items : SET [1:?] OF document_reference_item;
    END_ENTITY; -- applied_document_reference
(*)
applied_document_usage_constraint_assignment
*)
    ENTITY applied_document_usage_constraint_assignment
        SUBTYPE OF (document_usage_constraint_assignment);
        items : SET [1:?] OF document_reference_item;
    END_ENTITY; -- applied_document_usage_constraint_assignment
(*)
applied_effectivity_assignment
*)
    ENTITY applied_effectivity_assignment
        SUBTYPE OF (effectivity_assignment);
        items : SET [1:?] OF effectivity_item;
    END_ENTITY; -- applied_effectivity_assignment
(*)
applied_external_identification_assignment
*)
    ENTITY applied_external_identification_assignment
        SUPERTYPE OF (ONEOF (library_property_version_assignment,
            library_class_version_assignment))
        SUBTYPE OF (external_identification_assignment);
        items : SET [1:?] OF external_identification_item;
    END_ENTITY; -- applied_external_identification_assignment
(*)
applied_group_assignment
*)
    ENTITY applied_group_assignment
        SUBTYPE OF (group_assignment);
        items : SET [1:?] OF group_item;
    END_ENTITY; -- applied_group_assignment
(*)
applied_identification_assignment
*)
    ENTITY applied_identification_assignment
        SUBTYPE OF (identification_assignment);
        items : SET [1:?] OF identification_assignment_item;
    END_ENTITY; -- applied_identification_assignment
(*)
applied_library_assignment
*)
    ENTITY applied_library_assignment
        SUBTYPE OF (applied_classification_assignment);
        WHERE
    wr1: SELF\classification_assignment.role.name =
        'definitional class membership';
    wr2: (SELF\classification_assignment.assigned_class.name =
        'library identifier') AND (
        'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_CLASS' IN TYPEOF(SELF\
        classification_assignment.assigned_class));

```

```

wr3: SIZEOF(QUERY ( edir <* USEDIN(SELF.assigned_class,
'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATED_ITEM')
| ((edir.name = 'name scope') AND (
'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_GENERAL_PROPERTY' IN
TYPEOF(edir.relatering_item))) AND (SIZEOF(QUERY ( gpa <*
USEDIN(edir.relatering_item,
'CAST_PARTS_SCHEMA.GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION')
| ((gpa.name = 'definitional') AND (SIZEOF(QUERY ( pdr <*
USEDIN(gpa.derived_definition,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| ((pdr.used_representation.name = 'property value') AND (
SIZEOF(QUERY ( ri <* pdr.used_representation.items | (
'CAST_PARTS_SCHEMA.VALUE_REPRESENTATION_ITEM' IN TYPEOF(ri)) ))
= 1)) )) = 1)) )) = 1)) )) = 1;
END_ENTITY; -- applied_library_assignment

```

(\*

### **applied\_organization\_assignment**

\*)

```

ENTITY applied_organization_assignment
SUBTYPE OF (organization_assignment);
items : SET [1:?] OF organization_item;
END_ENTITY; -- applied_organization_assignment

```

(\*

### **applied\_person\_and\_organization\_assignment**

\*)

```

ENTITY applied_person_and_organization_assignment
SUBTYPE OF (person_and_organization_assignment);
items : SET [1:?] OF person_and_organization_item;
END_ENTITY; -- applied_person_and_organization_assignment

```

(\*

### **applied\_security\_classification\_assignment**

\*)

```

ENTITY applied_security_classification_assignment
SUBTYPE OF (security_classification_assignment);
items : SET [1:?] OF security_classification_item;
END_ENTITY; -- applied_security_classification_assignment

```

(\*

### **applied\_time\_assignment**

\*)

```

ENTITY applied_time_assignment
SUBTYPE OF (time_assignment);
items : SET [1:?] OF time_item;
END_ENTITY; -- applied_time_assignment

```

(\*

### **approval**

\*)

```

ENTITY approval;
status : approval_status;
level : label;
END_ENTITY; -- approval

```

(\*

**approval\_assignment**

```

*)
ENTITY approval_assignment
  ABSTRACT SUPERTYPE;
  assigned_approval : approval;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_assignment

```

(\*)

**approval\_date\_time**

```

*)
ENTITY approval_date_time;
  date_time      : date_time_select;
  dated_approval : approval;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_date_time

```

(\*)

**approval\_person\_organization**

```

*)
ENTITY approval_person_organization;
  person_organization : person_organization_select;
  authorized_approval : approval;
  role                : approval_role;
END_ENTITY; -- approval_person_organization

```

(\*)

**approval\_role**

```

*)
ENTITY approval_role;
  role : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
      'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- approval_role

```

(\*)

**approval\_status**

```

*)
ENTITY approval_status;
  name : label;
END_ENTITY; -- approval_status

```

(\*)

**assembly**

(\*)

## ISO 10303-223:2008 (E)

```
ENTITY assembly
  SUBTYPE OF (non_casting_activity);
  WHERE
```

```
(* ----- pattern section assembly sequence ----- *)
```

```
wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'pattern section assembly sequence')
  )))) >=0 ))))>=1;
```

```
(* ----- foam pattern coating density ----- *)
```

```
wr2:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'foam pattern coating density')
  )))) <=1 ))))>=1;
```

```
END_ENTITY; -- assembly
```

```
(*
```

### **assembly\_component\_usage**

```
*)
```

```
ENTITY assembly_component_usage
  SUPERTYPE OF (next_assembly_usage_occurrence)
  SUBTYPE OF (product_definition_usage);
  reference_designator : OPTIONAL identifier;
END_ENTITY; -- assembly_component_usage
```

```
(*
```

### **axis1\_placement**

```
*)
```

```
ENTITY axis1_placement
  SUBTYPE OF (placement);
  axis : OPTIONAL direction;
  DERIVE
  z : direction := NVL(normalise(axis), dummy_gri || direction([0,0,1]));
  WHERE
  wr1: SELF\geometric_representation_item.dim = 3;
END_ENTITY; -- axis1_placement
```

```
(*
```

### **axis2\_placement\_2d**

```
*)
```

```
ENTITY axis2_placement_2d
  SUBTYPE OF (placement);
  ref_direction : OPTIONAL direction;
  DERIVE
  p : LIST [2:2] OF direction := build_2axes(ref_direction);
```

```

WHERE
  wr1: SELF\geometric_representation_item.dim = 2;
END_ENTITY; -- axis2_placement_2d
(*)
axis2_placement_3d
*)
ENTITY axis2_placement_3d
  SUBTYPE OF (placement);
  axis          : OPTIONAL direction;
  ref_direction : OPTIONAL direction;
  DERIVE
  p : LIST [3:3] OF direction := build_axes(axis,ref_direction);
  WHERE
  wr1: SELF\placement.location.dim = 3;
  wr2: (NOT EXISTS(axis)) OR (axis.dim = 3);
  wr3: (NOT EXISTS(ref_direction)) OR (ref_direction.dim = 3);
  wr4: ((NOT EXISTS(axis)) OR (NOT EXISTS(ref_direction))) OR (
    cross_product(axis,ref_direction).magnitude > 0);
END_ENTITY; -- axis2_placement_3d
(*)
b_spline_curve
*)
ENTITY b_spline_curve
  SUPERTYPE OF (ONEOF (uniform_curve,b_spline_curve_with_knots,
    quasi_uniform_curve,bezier_curve) ANDOR rational_b_spline_curve)
  SUBTYPE OF (bounded_curve);
  degree          : INTEGER;
  control_points_list : LIST [2:?] OF cartesian_point;
  curve_form      : b_spline_curve_form;
  closed_curve    : LOGICAL;
  self_intersect  : LOGICAL;
  DERIVE
  upper_index_on_control_points : INTEGER := SIZEOF(
    control_points_list) - 1;
  control_points                : ARRAY [0:
    upper_index_on_control_points] OF
    cartesian_point := list_to_array(
    control_points_list,0,
    upper_index_on_control_points);
  WHERE
  wr1: (((('CAST_PARTS_SCHEMA.UNIFORM_CURVE' IN TYPEOF(SELF)) OR (
    'CAST_PARTS_SCHEMA.QUASI_UNIFORM_CURVE' IN TYPEOF(SELF))) OR
    ('CAST_PARTS_SCHEMA.BEZIER_CURVE' IN TYPEOF(SELF))) OR (
    'CAST_PARTS_SCHEMA.B_SPLINE_CURVE_WITH_KNOTS' IN
    TYPEOF(SELF)));
END_ENTITY; -- b_spline_curve
(*)

```

### **b\_spline\_curve\_with\_knots**

```

*)
ENTITY b_spline_curve_with_knots
  SUBTYPE OF (b_spline_curve);

```

```

    knot_multiplicities : LIST [2:?] OF INTEGER;
    knots                : LIST [2:?] OF parameter_value;
    knot_spec            : knot_type;
DERIVE
    upper_index_on_knots : INTEGER := SIZEOF(knots);
WHERE
    wr1: constraints_param_b_spline(degree, upper_index_on_knots,
        upper_index_on_control_points, knot_multiplicities, knots);
    wr2: SIZEOF(knot_multiplicities) = upper_index_on_knots;
END_ENTITY; -- b_spline_curve_with_knots
(*)

```

## **b\_spline\_surface**

```

*)
ENTITY b_spline_surface
    SUPERTYPE OF (ONEOF (b_spline_surface_with_knots, uniform_surface,
        quasi_uniform_surface, bezier_surface) ANDOR
        rational_b_spline_surface)
    SUBTYPE OF (bounded_surface);
    u_degree          : INTEGER;
    v_degree          : INTEGER;
    control_points_list : LIST [2:?] OF LIST [2:?] OF cartesian_point;
    surface_form      : b_spline_surface_form;
    u_closed          : LOGICAL;
    v_closed          : LOGICAL;
    self_intersect    : LOGICAL;
DERIVE
    u_upper          : INTEGER := SIZEOF(control_points_list) - 1;
    v_upper          : INTEGER := SIZEOF(control_points_list[1]) - 1;
    control_points   : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF
        cartesian_point := make_array_of_array(
            control_points_list, 0, u_upper, 0, v_upper);
WHERE
    wr1: ((('CAST_PARTS_SCHEMA.UNIFORM_SURFACE' IN TYPEOF(SELF)) OR (
        'CAST_PARTS_SCHEMA.QUASI_UNIFORM_SURFACE' IN TYPEOF(SELF)))
        OR ('CAST_PARTS_SCHEMA.BEZIER_SURFACE' IN TYPEOF(SELF))) OR
        ('CAST_PARTS_SCHEMA.B_SPLINE_SURFACE_WITH_KNOTS' IN TYPEOF(
            SELF));
END_ENTITY; -- b_spline_surface
(*)

```

## **b\_spline\_surface\_with\_knots**

```

*)
ENTITY b_spline_surface_with_knots
    SUBTYPE OF (b_spline_surface);
    u_multiplicities : LIST [2:?] OF INTEGER;
    v_multiplicities : LIST [2:?] OF INTEGER;
    u_knots          : LIST [2:?] OF parameter_value;
    v_knots          : LIST [2:?] OF parameter_value;
    knot_spec        : knot_type;
DERIVE
    knot_u_upper : INTEGER := SIZEOF(u_knots);
    knot_v_upper : INTEGER := SIZEOF(v_knots);
WHERE
    wr1: constraints_param_b_spline(SELF\b_spline_surface.u_degree,

```

```

        knot_u_upper, SELF\b_spline_surface.u_upper, u_multiplicities,
        u_knots);
    wr2: constraints_param_b_spline(SELF\b_spline_surface.v_degree,
        knot_v_upper, SELF\b_spline_surface.v_upper, v_multiplicities,
        v_knots);
    wr3: SIZEOF(u_multiplicities) = knot_u_upper;
    wr4: SIZEOF(v_multiplicities) = knot_v_upper;
END_ENTITY; -- b_spline_surface_with_knots
(*)
bankout_frame
*)
ENTITY bankout_frame
    SUBTYPE OF (core_equipment);
    WHERE
        wr1: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
            (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
            TYPEOF(pdr.representation)) AND
            (SIZEOF(QUERY (it <* pdr.representation.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND ((it.name = 'material type')
            )))) =1 ))))>=1;

    END_ENTITY; -- bankout_frame
(*)
bench_inspection
*)
ENTITY bench_inspection
    SUBTYPE OF (inspection_activity);
END_ENTITY; -- bench_inspection
(*)
bezier_curve
*)
ENTITY bezier_curve
    SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve
(*)
bezier_surface
*)
ENTITY bezier_surface
    SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface
(*)
boss
*)
ENTITY boss
    SUBTYPE OF (feature_definition);
    WHERE
        wr1 : SELF\characterized_object.description IN ['circular', 'complex',
            'rectangular'];
        wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (

```

```

'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'boss height occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND (('CAST_PARTS_SCHEMA.'
+ 'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (((
'CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(sdr.
relating_shape_aspect)) AND (sdr.relating_shape_aspect.
description = 'linear')) AND (sdr.name = 'boss height')) )
= 1)) = 1)) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND ((1 <= SIZEOF(pdr.
used_representation.items)) AND (SIZEOF(pdr.
used_representation.items) <= 2))) ) = 1)) = 1;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
used_representation.items | ((srwp_i.name = 'orientation')
OR (srwp_i.name = 'fillet radius')) ) = SIZEOF(pdr.
used_representation.items))) ) = 1)) = 1;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'fillet radius')) ) <= 1)) ) = 0)) )
= 0;
wr6 : (NOT (SELF\characterized_object.description = 'circular')) OR
(SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'circular profile occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,

```



```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relatering_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr7 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'top condition occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
| ((sar.description = 'boss top usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | ('CAST_PARTS_SCHEMA.BOSS_TOP' IN TYPEOF(
fcr.relatering_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr8 : (NOT (SELF\characterized_object.description = 'circular')) OR
(SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'change in diameter occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
AND ('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(fcr
.relatering_shape_aspect)) AND ('CAST_PARTS_SCHEMA.BOSS' IN
TYPEOF(fcr.relatering_shape_aspect))) )) = 1)) )) <= 1)) )) =
0);
wr9 : (NOT (SELF\characterized_object.description = 'complex')) OR (
SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'enclosed boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (SIZEOF(['CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
relatering_shape_aspect)) = 1)) )) = 1)) )) = 1)) )) = 0);
wr10: (NOT (SELF\characterized_object.description IN ['complex',
'rectangular'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )

```

```

    | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'change in boundary occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | ((sar.description =
    'taper usage') AND (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(
fcr.related_shape_aspect)) AND ('CAST_PARTS_SCHEMA.BOSS' IN
    TYPEOF(fcr.relater_shape_aspect))) AND (fcr.
related_shape_aspect.description IN ['angle taper',
'directed taper'])) ) = 1)) ) <= 1)) ) = 0);
wr11: (NOT (SELF\characterized_object.description = 'rectangular'))
OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
    | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'rectangular profile occurrence') AND (
SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | ((sar.description =
    'profile usage') AND (
    'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
IN TYPEOF(sdr.relater_shape_aspect)) ) = 1)) ) = 1)) )
= 0);
wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit')) ) ) >=
0;
END_ENTITY; -- boss
(*)
boss_top
*)
ENTITY boss_top
SUBTYPE OF (shape_aspect);
WHERE
wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
.of_shape.definition);
wr2: SELF.description IN ['planar','complex'];
wr3: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
    TYPEOF(pdr.used_representation)) ) = 1)) ) = 0);
wr4: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (

```

```

SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
'CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(
pdr.used_representation)) )) = 1)) )) = 0);
wr5: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr6: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | (((sar.description =
'boss top usage') AND (sar.name IN ['boss height start',
'boss height end'])) AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
'top condition occurrence') AND ('CAST_PARTS_SCHEMA.BOSS' IN
TYPEOF(fcr.related_shape_aspect.of_shape.definition))) AND
('CAST_PARTS_SCHEMA.BOSS_TOP' IN TYPEOF(fcr.
relating_shape_aspect)))) >= 1;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) <= 1)) )) = 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1)) )) = 0)) )) = 0;
wr9: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'top radius')) )) <= 1)) )) = 0)) )) = 0;
END_ENTITY; -- boss_top
(*

```

## boundary\_condition

© ISO 2008 — All rights reserved

1435

```

*)
ENTITY boundary_condition
  SUBTYPE OF (action_method);
  WHERE
  wr1: SIZEOF(QUERY ( ama <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')
    | (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN
    TYPEOF(ama)) AND (NOT (SIZEOF(QUERY ( eds <* ama.items | ((
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    (SIZEOF(QUERY ( pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
    | (SIZEOF(QUERY ( mri <* pd.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP'] * TYPEOF(mri)) =
    1) AND (mri.name = 'condition value')) ) = 1) ) = 1) ) =
    1))) ) = 0;
  wr2: NOT (SIZEOF(QUERY ( ama <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')
    | (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN
    TYPEOF(ama)) AND (SIZEOF(QUERY ( eds <* ama.items | (
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(eds)) ) = 1) ) )
    = 0);
  END_ENTITY; -- boundary_condition

```

(\*

## bounded\_curve

\*)

```

ENTITY bounded_curve
  SUPERTYPE OF (ONEOF (polyline,b_spline_curve,composite_curve))
  SUBTYPE OF (curve);
  END_ENTITY; -- bounded_curve

```

(\*

## bounded\_surface

\*)

```

ENTITY bounded_surface
  SUPERTYPE OF (b_spline_surface)
  SUBTYPE OF (surface);
  END_ENTITY; -- bounded_surface

```

(\*

## brep\_with\_voids

\*)

```

ENTITY brep_with_voids
  SUBTYPE OF (manifold_solid_brep);
  voids : SET [1:?] OF oriented_closed_shell;
  END_ENTITY; -- brep_with_voids

```

(\*

## calendar\_date

\*)

```

ENTITY calendar_date
  SUBTYPE OF (date);
  day_component : day_in_month_number;
  month_component : month_in_year_number;
  WHERE

```

```

        wr1: valid_calendar_date(SELF);
    END_ENTITY; -- calendar_date
(*)
cartesian_point
*)
    ENTITY cartesian_point
        SUBTYPE OF (point);
        coordinates : LIST [1:3] OF length_measure;
    END_ENTITY; -- cartesian_point
(*)
cartesian_transformation_operator
*)
    ENTITY cartesian_transformation_operator
        SUPERTYPE OF (cartesian_transformation_operator_3d)
        SUBTYPE OF (geometric_representation_item,
            functionally_defined_transformation);
        axis1      : OPTIONAL direction;
        axis2      : OPTIONAL direction;
        local_origin : cartesian_point;
        scale      : OPTIONAL REAL;
    DERIVE
        scl : REAL := NVL(scale,1);
    WHERE
        wr1: scl > 0;
    END_ENTITY; -- cartesian_transformation_operator
(*)
cartesian_transformation_operator_3d
*)
    ENTITY cartesian_transformation_operator_3d
        SUBTYPE OF (cartesian_transformation_operator);
        axis3 : OPTIONAL direction;
    DERIVE
        u : LIST [3:3] OF direction := base_axis(3,SELF\
            cartesian_transformation_operator.axis1,SELF\
            cartesian_transformation_operator.axis2,axis3);
    WHERE
        wr1: SELF\geometric_representation_item.dim = 3;
    END_ENTITY; -- cartesian_transformation_operator_3d
(*)
cast_part
*)
    ENTITY cast_part
        SUBTYPE OF (product_definition);
    WHERE
        wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
            ( 'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE' IN TYPEOF (sa)
            )))>=1;

    END_ENTITY; -- cast_part
(*)
cast_part_with_rigging

```

```

*)
ENTITY cast_part_with_rigging
  SUBTYPE OF (product_definition);
  WHERE
wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ( 'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE' IN TYPEOF (sa)
  )))>=1;

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  (SIZEOF([
  'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE',
  'CAST_PARTS_SCHEMA.GATING_SYSTEM'] * TYPEOF(sa) = 2)))=1;

END_ENTITY; -- cast_part_with_rigging
(*
casting_activity
*)
ENTITY casting_activity
  SUBTYPE OF (process_plan_activity);
  WHERE
wr1: SIZEOF(QUERY ( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')
  | (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN
  TYPEOF(ama)) AND (SIZEOF(QUERY ( eds <* ama.items | (
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA' IN TYPEOF(eds)) )
  <= 1)) ) ) <= 1;
END_ENTITY; -- casting_activity
(*
casting_design_feature
*)
ENTITY casting_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE
wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  ('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) ) ) = 1;

(* ***** casting_design_feature to orientation (as placement) ***** *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) = 1 ))))=1;

(* ----- draft ----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',

```

```

        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
    (it.name = 'draft')) )) =1 ))))=1;

(* ***** machine_stock ***** *)

wr4: SIZEOF(QUERY ( co <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    ('CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF(co)) )) <= 1;

    END_ENTITY; -- casting_design_feature
(*
casting_envelope_requirement
*)
ENTITY
    SUBTYPE OF (property_definition);
    WHERE

    (* ----- to numeric_parameter(as part_height) -----*)

wr1:(NOT (SIZEOF(QUERY (pdr <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) AND
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
    (it.name = 'part height')) )) =0 )) )) =0));

    (* ----- to numeric_parameter(as part_length) -----*)

wr2:(NOT (SIZEOF(QUERY (pdr <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) AND
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
    (it.name = 'part length')) )) =0 )) )) =0));

    (* ----- to numeric_parameter(as part_width) -----*)

wr3:(NOT (SIZEOF(QUERY (pdr <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(pdr.used_representation)) AND
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
    (it.name = 'part width')) )) =0 )) )) =0));

    END_ENTITY; -- casting_envelope_requirement
(*

```

## casting\_equipment

```

*)
  ENTITY casting_equipment
    SUPERTYPE OF (ONEOF (core_equipment,ejector_equipment,shakeout,
      finishing_equipment,moulding_equipment,melting_equipment))
    SUBTYPE OF (machine);
  WHERE

(* ---- equipment_name ---- *)

wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.ORGANIZATION'
  IN TYPEOF(adr.assigned_organization)) AND
  (adr.assigned_organization.description = 'equipment company name')
  ) ) ) = 1;

(* ---- equipment_type---- *)

wr2:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'equipment type')
  ))) = 1 ))) >= 1;

(* ---- equipment to controller (as controlled_by) ---- *)

wr3: SIZEOF(QUERY ( arr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE') |
  (('CAST_PARTS_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP' IN TYPEOF(arr)) AND
  ('CAST_PARTS_SCHEMA.CONTROLLER' IN TYPEOF(arr.relating_resource)
  ))) = 1;

(* ---- equipment to in_facility_location (as location) ---- *)

wr4: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION' IN
  TYPEOF(pdr.representation)) AND
  (pdr.representation.name='location')) ) <= 1;

  END_ENTITY; -- equipment
(*

```

## casting\_feature\_definition

```

*)
  ENTITY casting_feature_definition
    SUPERTYPE OF (ONEOF (sand_cast_design_feature,die_design_feature,
      ejector_design_feature,gating_design_feature,casting_design_feature,
      investment_design_feature))
    SUBTYPE OF (characterized_object);
  WHERE
    wr1: SIZEOF(['CAST_PARTS_SCHEMA.EJECTOR_DESIGN_FEATURE',

```



```

        'CAST_PARTS_SCHEMA.CASTING_DESIGN_FEATURE',
        'CAST_PARTS_SCHEMA.GATING_DESIGN_FEATURE',
        'CAST_PARTS_SCHEMA.DIE_DESIGN_FEATURE',
        'CAST_PARTS_SCHEMA.INVESTMENT_DESIGN_FEATURE',
        'CAST_PARTS_SCHEMA.SAND_CAST_DESIGN_FEATURE'] * TYPEOF(SELF))
        <= 1;
    END_ENTITY; -- casting_feature_definition
(*
casting_feature_size
*)
    ENTITY casting_feature_size
        SUBTYPE OF (shape_aspect);
        WHERE
(* ----- item_size to numeric_parameter (as item_height)----- *)

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'item height')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_length)----- *)

    wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'item length')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_width)----- *)

    wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'item width')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_diameter)----- *)

    wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',

```

## ISO 10303-223:2008 (E)

```
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it) = 2) AND
    (it.name = 'item diameter')) ) <=1 )))=1;

(* ----- item_size to numeric_parameter (as item_draft)----- *)

wr5:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it) = 2) AND
    (it.name = 'item draft')) ) <=1 ))))=1;

    END_ENTITY; -- casting_feature_size
(*
casting_insert
*)
    ENTITY casting_insert
        SUBTYPE OF (sand_cast_design_feature);
        WHERE
(* ***** insert_name ***** *)

    wr1 :   SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF)
    |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
            (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'insert name')))) =1 ))))=1;

(* ***** material_type ***** *)

    wr2:   (SIZEOF(QUERY (co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

(* ***** positioning_method ***** *)

    wr3 :   SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
            (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'positioning method')))) =1 ))))=1;

    END_ENTITY; -- casting_insert
(*
casting_instanced_feature
*)
    ENTITY casting_instanced_feature
        SUBTYPE OF (casting_feature_definition, shape_aspect);
1442
```

© ISO 2008—All rights reserved

```

WHERE
  wr1: 'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(SELF.of_shape.
        definition);
  wr2: SELF.product_definitional;
END_ENTITY; -- casting_instanced_feature
(*)
casting_method
*)
ENTITY casting_method
  SUBTYPE OF (action_method);
END_ENTITY; -- casting_method
(*)
casting_per_order
*)
ENTITY casting_per_order
  SUBTYPE OF (casting_verification);
WHERE
  wr1: NOT (SIZEOF(QUERY ( ad <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_DIRECTIVE.REQUESTS') | (NOT (
      SIZEOF(QUERY ( da <* USEDIN(ad,
        'CAST_PARTS_SCHEMA.DIRECTED_ACTION.DIRECTIVE') | ((
          'CAST_PARTS_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(da)) AND (
            SIZEOF(QUERY ( ppa <* USEDIN(da,
              'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
                ('CAST_PARTS_SCHEMA.SAMPLED_SET' IN TYPEOF(ppa.
                  property_or_shape)) ) = 1)) ) = 0)) ) = 0);
END_ENTITY; -- casting_per_order
(*)
casting_process
*)
ENTITY casting_process
  SUBTYPE OF (casting_activity);
WHERE

(*) ----- core_making_process ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'core making process') AND
    (it.description IN ['shell','no bake','lost form','gas cured',
      'green sand']))) = 1 )))) >= 1;

wr2: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'

```

```

    IN TYPEOF(it)) AND (it.name = 'core making process') AND
    (it.description IN ['core presentation',
'core setting',
'die assembly',
'die handling',
'die temperature control',
'furnace loading',
'gate system removal',
'grinding',
'inspection',
'ladle loading',
'loading injection system',
'melt raw material',
'mould closing',
'mould making process',
'package and shipping',
'part routing',
'pattern handling',
'tool handling',
'pouring process',
'refractor removal',
'removal from die',
'shakeout',
'slurry process',
'sand removal',
'wax removal',
'shell',
'green sand',
'no bake',
'lost form',
'vacuum',
'static',
'tilt',
'low pressure'])) =1 ))))>=1;

```

```

wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
(('CAST_PARTS_SCHEMA.DEFINING_ACTION_METHOD_RELATIONSHIP'
IN TYPEOF(ap)) AND
('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT_SETUP'
IN TYPEOF(ap.related_method))))=1;

```

(\* ----- casting\_process to mismatch\_tolerance (as tolerances)----- \*)

```

wr4: (SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.MISMATCH_TOLERANCE' IN TYPEOF (eds))
)) <=1 )) )<=1);

```

```
(* --- casting_process to production_core_box (as core_to_be_made)---- *)

wr5: (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX' IN TYPEOF (eds.definition))
  )) =1 )) ) ) =0);

(* ---- casting_process to numeric_parameter (as process_parameter) ----- *)

wr6:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.representation))
AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP'] * TYPEOF
  (it)) = 1) AND (it.name = 'special instruction'))
  )) =1 ))) >=1;

(* ---- casting_process to shape_aspect (as described_by) ----- *)

wr7: (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
  )) <=1 )) ) >=0);

  END_ENTITY; -- casting_process
(*
casting_product_definition
*)
  ENTITY casting_product_definition
    SUBTYPE OF (product_definition);
  WHERE
  WR1: 'CAST_PARTS_SCHEMA.APPLICATION_CONTEXT_ELEMENT'
    IN TYPEOF(SELF.frame_of_reference);

  WR2: SELF.frame_of_reference.name = 'function definition';

(* --- tooling to cast_part (as tooling designed_for)---- *)

wr3:  SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CAST_PART'
  IN TYPEOF(pdr.relatng_product_definition))))=1;
```

ISO 10303-223:2008 (E)

```
(* --- tooliong to date ( as date_last_used)---- *)

wr4: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
    (ada.role.name='date last used'))=1);
END_ENTITY; -- casting_product_definition
(*
casting_service_data
*)
ENTITY casting_service_data
    SUBTYPE OF (customer_casting_requirement);
    WHERE

(* ----- to corrosion_service (as corrosion_service_requirement)----- *)

wr1:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.CORROSION_SERVICE'
    IN TYPEOF(pdr.related_property_definition)
    ))>=0);

(* ----- to descriptive_parameter(as comments) ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'comments')
    ) ) =1 ))=0));

(* ----- wear_with_lubrication ----- *)

wr3:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'wear or abrasion lubrication') AND
    (it.description IN ['good','intermitten','none'] )
    ) )=1 )))) = 0);

(* -- to elevated_temperature_service (as elevated_temperature_service) *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    ('CAST_PARTS_SCHEMA.ELEVATED_TEMPERATURE_SERVICE'
    IN TYPEOF(pdr.related_property_definition)
    ))=1);

(* finishing_or_machining_operations(machining_and_finishing_requirements) *)

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
```

```

'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.FINISHING_AND_MACHINING_OPERATION'
 IN TYPEOF(pdr.related_property_definition)
)=1);

(* ----- to material (as subject_to_abrasive_wear_by)----- *)

wr6:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'subject to abrasize wear against')
) )) =1 )))=0));

(* ----- to material (as subject_to_wear_against)----- *)

wr7:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'subject to wear against')
) )) =1 )))=0));

(* --- to mechanical_stress (as mechanical_stress_requirements) ----- *)

wr8:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.MECHANICAL_STRESS'
 IN TYPEOF(pdr.related_property_definition)
)=1);

END_ENTITY; -- casting_service_data
(*
casting_verification
*)
ENTITY casting_verification
  SUBTYPE OF (versioned_action_request);
  WHERE

  (* ----- cavity_impression_id -----*)

wr1:(NOT(SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
 IN TYPEOF(it)) AND

```

ISO 10303-223:2008 (E)

```
(it.name = 'cavity impression id'))
)) =1 ))) =0 )))) =0 )))) =0 )));

(* --- casting_verification to tolerance_requirement(as verify_with) --- *)

wr2: SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT(SIZEOF(QUERY(ari <* ada.items |
  'CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
  IN TYPEOF(ari))) =1))))=0;

(* ----- to quality_acceptance_report (as destruction) ----- *)

wr3: SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT(SIZEOF(QUERY(ari <* ada.items |
  (('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF(ari)) AND
  (ari.kind.product_data_type='quality acceptance report') AND
  (ari\document.description='destruction'))
  )) =1))))=0;

(* ----- to quality_acceptance_report (as nde) ----- *)

wr4: SIZEOF(QUERY(ada <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF(ada)) AND
  (NOT(SIZEOF(QUERY(ari <* ada.items |
  (('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF(ari)) AND
  (ari.kind.product_data_type='quality acceptance report') AND
  (ari\document.description='nde'))
  )) =1))))=0;

END_ENTITY; -- casting_verification
(*
centre_of_symmetry
*)
ENTITY centre_of_symmetry
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    wr1: SIZEOF(QUERY ( sadr <* deriving_relationships | (NOT (
      'CAST_PARTS_SCHEMA.SYMMETRIC_SHAPE_ASPECT' IN TYPEOF(sadr.
      related_shape_aspect))) ) ) = 0;
END_ENTITY; -- centre_of_symmetry
(*
chamfer
*)
ENTITY chamfer
```



```

SUBTYPE OF (transition_feature);
WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
    used_representation)) AND (pdr.used_representation.name =
    'chamfer face')) ) <= 1)) ) = 0;
  wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) ) | ((( 'CAST_PARTS_SCHEMA.CHAMFER_OFFSET' IN
    TYPEOF(fcr.related_shape_aspect)) AND (
    'CAST_PARTS_SCHEMA.CHAMFER' IN TYPEOF(fcr.
    relating_shape_aspect))) AND (fcr.related_shape_aspect.
    description = 'first offset')) ) = 1;
  wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) ) | ((( 'CAST_PARTS_SCHEMA.CHAMFER_OFFSET' IN
    TYPEOF(fcr.related_shape_aspect)) AND (
    'CAST_PARTS_SCHEMA.CHAMFER' IN TYPEOF(fcr.
    relating_shape_aspect))) AND (fcr.related_shape_aspect.
    description = 'second offset')) ) = 1;
END_ENTITY; -- chamfer

```

(\*

## chamfer\_offset

```

*)
ENTITY chamfer_offset
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: SELF.description IN ['first offset', 'second offset'];
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) = 1)) ) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 1)) ) = 0)) ) = 0;
    wr4: (NOT (SELF.description = 'first offset')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'offset amount')) ) = 1)) ) = 0)) ) =
0);
wr5: (NOT (SELF.description = 'first offset')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'first face shape')) ) <= 1)) ) = 0);
wr6: (NOT (SELF.description = 'second offset')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'offset amount')) OR ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
it)) = 2) AND (it.name = 'offset angle')) ) = 1)) ) = 0)) ) =
0);
wr7: (NOT (SELF.description = 'second offset')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'second face shape')) ) <= 1)) ) = 0);
wr8: SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | (('CAST_PARTS_SCHEMA.' +
'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) ) | ((
'CAST_PARTS_SCHEMA.CHAMFER' IN TYPEOF(sdr.
relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.CHAMFER_OFFSET' IN TYPEOF(sdr.
related_shape_aspect))) ) = 1;

```

```

    END_ENTITY; -- chamfer_offset
  (*
chaplet
  *)
    ENTITY chaplet
      SUBTYPE OF (sand_cast_design_feature);
      WHERE
  (* ***** chaplet_type ***** *)

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'chaplet type')))) =1 ))))=1;

  (* ----- chaplet to item_size (as chaplet_size) ----- *)

    wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
      ((sa.description='chaplet reference occurrence') AND
      (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
      AND
      (sar.description = 'chaplet dimension usage') AND
      ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
      IN TYPEOF(sar.relatng_shape_aspect))
      )) >=1 ))) >=1;

  (* ***** chaplet to chaplet_pad (as pad) ***** *)

    wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
      ((sa.description='chaplet reference occurrence') AND
      (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
      (sar.description = 'chaplet pad reference usage') AND
      ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relatng_shape_aspect))
      )) <=1 ))) >=1;

    END_ENTITY; -- chaplet
  (*
chaplet_pad
  *)
    ENTITY chaplet_pad
      SUBTYPE OF (sand_cast_design_feature);
      WHERE
    wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
      ((sa.description='chaplet pad dimension occurrence') AND
      (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP')
      IN TYPEOF(sar)) AND

```

```

        (sar.description = 'chaplet pad dimension usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
        TYPEOF(sar.relatng_shape_aspect))
        )) =1 )))) =1;
    END_ENTITY; -- chaplet_pad
    (*
characterized_object
    *)
    ENTITY characterized_object;
        name      : label;
        description : OPTIONAL text;
    END_ENTITY; -- characterized_object
    (*
checking_aid_tool
    *)
    ENTITY checking_aid_tool
        SUBTYPE OF (customer_casting_requirement);
        WHERE
    (* ----- aid id ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'aid id')
    ) ))=1 )))) = 0);

    (* ----- aid name----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'aid name')
    ) ))=1 )))) = 0);

    (* ----- aid_tool ----- *)

wr3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'aid tool') AND
    (it.description IN ['fixture','guage','paper document','template'] )
    ) ))=1 )))) = 0);

    (* ----- revision id ----- *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```

```

        (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'revision id')
        ) )=1 ))) = 0);

(* ---      checking_aid_tool to date ( as revision_date)----- *)

wr5: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
        (ada.role.name='revision date')))=1);

    END_ENTITY; -- checking_aid_tool
(*
chill
*)
    ENTITY chill
        SUBTYPE OF (sand_cast_design_feature);
        WHERE
(* -----      internal_or_external_chill----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND ((it.name = 'internal or external chill') AND
        ((it.description = 'true') OR (it.description = 'false'))
        )))) =1 )))>=1;

(* -----      material_type ----- *)

wr2: SIZEOF(QUERY ( co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        ('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) ) ) =1;

(* -----      chill to shape_element (as chill_area_shape)----- *)

wr3:  SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='chill area occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'chill area shape usage') AND
        ('CAST_PARTS_SCHEMA.SHAPE_ASPECT'
        IN TYPEOF(sar.relatng_shape_aspect))
        ) ) >=1 ))) =1;

    END_ENTITY; -- chill
(*
choke
*)
    © ISO 2008 — All rights reserved

```

ISO 10303-223:2008 (E)

```
ENTITY choke
  SUBTYPE OF (gating_design_feature);
  WHERE

(* ----- choke to item_size (as core_vent_size) ----- *)

WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='choke occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'choke dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

(* ----- choke to location_element (as vent_placement)----- *)

wr2 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='location')))=1;
```

```
END_ENTITY; -- choke
```

```
(*
```

### **circle**

```
*)
```

```
ENTITY circle
  SUBTYPE OF (conic);
  radius : positive_length_measure;
END_ENTITY; -- circle
```

```
(*
```

### **circular\_closed\_profile**

```
*)
```

```
ENTITY circular_closed_profile
  SUBTYPE OF (shape_aspect);
  WHERE
  wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
    .of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) = 1)) ) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```

used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 2)) ) = 0)) ) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) ) = 1)) ) = 0)) ) = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'diameter')) ) = 1)) ) = 0)) ) = 0;
END_ENTITY; -- circular_closed_profile
(*)
circular_pattern
*)
ENTITY circular_pattern
SUBTYPE OF (replicate_feature);
WHERE
wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN
TYPEOF(sdr.related_shape_aspect)) ) = 1) ) <= 3)) ) = 0;
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) = 1)) ) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT ((SIZEOF(impl_rep.
used_representation.items) >= 3) AND (SIZEOF(impl_rep.
used_representation.items) <= 5))) ) = 0)) ) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'diameter'))) <= 1))) ) = 0)) ) = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
it)) = 2) AND (it.name = 'base feature rotation'))) <= 1))) )
= 0)) ) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component))) AND (it.name =
'number of features'))) ) = 1))) ) = 0)) ) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*

```



```

impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
it)) = 2) AND (it.name = 'angular_spacing')) = 1)) = 0)) = 0;
= 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) = 1)) = 0)) = 0)) = 0;
END_ENTITY; -- circular_pattern
(*)
circular_runout_tolerance
*)
ENTITY circular_runout_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) <= 2;
END_ENTITY; -- circular_runout_tolerance
(*)
class
*)
ENTITY class
  SUBTYPE OF (group);
END_ENTITY; -- class
(*)
classification_assignment
*)
ENTITY classification_assignment
  ABSTRACT SUPERTYPE;
  assigned_class : group;
  role           : classification_role;
END_ENTITY; -- classification_assignment
(*)
classification_role
*)
ENTITY classification_role;
  name           : label;
  description    : OPTIONAL text;
END_ENTITY; -- classification_role
(*)
closed_path_profile
*)
ENTITY closed_path_profile

```

```

SUBTYPE OF (shape_aspect);
WHERE
  wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
    .of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') | (
    NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) | (NOT (SIZEOF(impl_rep.
    used_representation.items) = 1)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((
    'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
    'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') | (
    NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0;
END_ENTITY; -- closed_path_profile
(*)
closed_shell
*)
ENTITY closed_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell
(*)
coaxiality_tolerance
*)
ENTITY coaxiality_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
WHERE
  wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
    datum_system) <= 2;
END_ENTITY; -- coaxiality_tolerance

```

(\*)

**common\_datum**

\*)

```

ENTITY common_datum
  SUBTYPE OF (composite_shape_aspect, datum);
  WHERE
    wr1: SIZEOF(SELF.component_relationships) = 2;
    wr2: SIZEOF(QUERY ( sar <* SELF.component_relationships | (NOT ((
      'CAST_PARTS_SCHEMA.DATUM' IN TYPEOF(sar.related_shape_aspect))
      AND (NOT ('CAST_PARTS_SCHEMA.COMMON_DATUM' IN TYPEOF(sar.
        related_shape_aspect)))))) ) = 0;
  END_ENTITY; -- common_datum

```

(\*)

**composite\_curve**

\*)

```

ENTITY composite_curve

  SUBTYPE OF (bounded_curve);
  segments      : LIST [1:?] OF composite_curve_segment;
  self_intersect : LOGICAL;
  DERIVE
    n_segments   : INTEGER := SIZEOF(segments);
    closed_curve : LOGICAL := segments[n_segments].transition <>
      discontinuous;
  WHERE
    wr1: ((NOT closed_curve) AND (SIZEOF(QUERY ( temp <* segments | (
      temp.transition = discontinuous) )) = 1)) OR (closed_curve
      AND (SIZEOF(QUERY ( temp <* segments | (temp.transition =
      discontinuous) )) = 0));
  END_ENTITY; -- composite_curve

```

(\*)

**composite\_curve\_on\_surface**

\*)

```

ENTITY composite_curve_on_surface
  SUBTYPE OF (composite_curve);
  DERIVE
    basis_surface : SET [0:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: SIZEOF(basis_surface) > 0;
    wr2: constraints_composite_curve_on_surface(SELF);
  END_ENTITY; -- composite_curve_on_surface

```

(\*)

**composite\_curve\_segment**

\*)

```

ENTITY composite_curve_segment
  SUBTYPE OF (founded_item);
  transition   : transition_code;
  same_sense   : BOOLEAN;
  parent_curve : curve;
  INVERSE
    using_curves : BAG [1:?] OF composite_curve FOR segments;
  WHERE

```

```

        wr1: 'CAST_PARTS_SCHEMA.BOUNDED_CURVE' IN TYPEOF(parent_curve);
    END_ENTITY; -- composite_curve_segment
    (*
composite_hole
    *)
    ENTITY composite_hole
        SUBTYPE OF (compound_feature);
    WHERE
        wr1: SELF\characterized_object.description IN ['counterbore',
            'countersunk'];
        wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
            'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
            AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
            AND (SIZEOF(QUERY ( sar <* csa.component_relationships | ((
            'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(sar)) AND ('CAST_PARTS_SCHEMA.ROUND_HOLE' IN TYPEOF(
            sar.related_shape_aspect))) = 2))) = 1))) = 1);
        wr3: (NOT (SELF\characterized_object.description = 'countersunk'))
            OR (SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
            'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
            AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
            'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
            AND (SIZEOF(QUERY ( sar <* csa.component_relationships | ((
            'CAST_PARTS_SCHEMA.ROUND_HOLE' IN TYPEOF(sar.
            related_shape_aspect)) AND (NOT (SIZEOF(QUERY ( pds <*
            QUERY ( pd <* USEDIN(sar.related_shape_aspect,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
            'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd))
            | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
            description = 'change in diameter occurrence') AND (SIZEOF(
            QUERY ( fcr2 <* QUERY ( sar2 <* USEDIN(sa_occ,
            'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
            | ((sar2.description = 'taper usage') AND (
            'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
            TYPEOF(sar2))) ) | ('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(fcr2
            .related_shape_aspect)) ) = 1))) = 0)) ) = 0))) ) = 1))) )
            = 1)) ) = 1);
    END_ENTITY; -- composite_hole
    (*

```

**composite\_shape\_aspect**

```

    *)
    ENTITY composite_shape_aspect
        SUBTYPE OF (shape_aspect);
    INVERSE
        component_relationships : SET [2:?] OF shape_aspect_relationship FOR
            relating_shape_aspect;
    END_ENTITY; -- composite_shape_aspect

```

(\*)

**composition\_element**

\*)

```
ENTITY composition_element
  SUBTYPE OF (material_property);
  WHERE
    (* --- element----- *)
```

```
WR1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(dri <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(dri)) AND
  (dri.name = 'element'))
  ))=1)))=0));
```

```
(* --- compositon_element to numeric_range (as amount)----- *)
```

```
WR2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY(dri <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT',
  'CAST_PARTS_SCHEMA.QUALIFIED_REPRESENTATION_ITEM']
  * TYPEOF(dri)) = 3) AND (dri.name = 'amount')) ) ) =1)))=0));
```

```
END_ENTITY; -- composition_element
```

(\*)

**composition\_element\_record**

\*)

```
ENTITY composition_element_record
  SUBTYPE OF (activity_result_record);
  WHERE
    (* -- to numeric_parameter (as actual_composition_amount) -----*)
```

```
wr1: (SIZEOF(QUERY( it <* SELF.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
  (it.name = 'actual_composition amount'))=1);
```

```
(* -----to compositon_element (as recorded_usage) -----*)
```

```
wr2: (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
  ('CAST_PARTS_SCHEMA.COMPOSITION_ELEMENT'
  IN TYPEOF(pdr.definition))))=1);
```

```
END_ENTITY; -- composition_element_record
```

(\*)

**composition\_inspection**

\*)

ISO 10303-223:2008 (E)

```
ENTITY composition_inspection
  SUBTYPE OF (inspection_or_test_result);
  WHERE
    (* ----- to material (as composition_result) -----*)

wr1: ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (SELF.definition));

(* -----to material_property (as requirements)----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.MATERIAL_PROPERTY'
  IN TYPEOF(pdr.related_property_definition)
  )))=1);

END_ENTITY; -- composition_inspection
(*
```

**compound\_feature**

```
*)
ENTITY compound_feature
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pds <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
      AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((csa.name =
      'compound feature in solid') AND (
      'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))) )
      = 1)) ) = 1;
    wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
      AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
      'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa)) )
      = 1)) ) = 1;
    wr3: SIZEOF(QUERY ( pds <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
      AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
      'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
      AND (SIZEOF(QUERY ( fcr <* csa.component_relationships | (
      NOT ('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(fcr))) ) = 0)) ) = 1)) ) = 1;
    wr4: SIZEOF(QUERY ( pds <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
      AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
```

```

        'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa)) ))
        = 1)) )) = 1;
wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
        AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
        AND (SIZEOF(QUERY ( sar <* csa.component_relationships | (
        'CAST_PARTS_SCHEMA.THREAD' IN TYPEOF(sar.
        related_shape_aspect)) )) = 0)) )) = 1)) )) = 1;
wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
        AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
        AND (SIZEOF(QUERY ( sar <* csa.component_relationships | ((
        'CAST_PARTS_SCHEMA.COMPOUND_FEATURE' IN TYPEOF(sar.
        related_shape_aspect)) AND (sar.related_shape_aspect\
        characterized_object.name <> SELF\characterized_object.name)) ))
        = 0)) )) = 1)) )) = 1;
END_ENTITY; -- compound_feature
(*)
compound_representation_item
*)
ENTITY compound_representation_item
    SUBTYPE OF (representation_item);
    item_element : compound_item_definition;
END_ENTITY; -- compound_representation_item
(*)
concentricity_tolerance
*)
ENTITY concentricity_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
        wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
            datum_system) = 1;
END_ENTITY; -- concentricity_tolerance
(*)
condition_or_assumption
*)
ENTITY condition_or_assumption
    SUBTYPE OF (action_method);
    WHERE
(*) condition_or_assumption to numeric_parameter(as associated_property) *)
wr1:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
        (NOT (SIZEOF(QUERY(eds <* ama.items |
        (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND

```

```

        (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (SIZEOF(QUERY(mri <* pd.used_representation.items |
        ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri))))=1)
        ))=1))))=0))))=0));

    (* to property_relationship(as associated_property) *)

wr2:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
    (NOT (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY(mri <* pd.used_representation.items |
    ('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN TYPEOF(mri))))=1)
    ))=1))))=0))))=0));

(* condition_or_assumption to CAST_PARTS_SCHEMA (as further_information) *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    ('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
    (adr.assigned_document))))=1;

    END_ENTITY; -- condition_or_assumption
    (*
conic
    *)
    ENTITY conic
        SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
        SUBTYPE OF (curve);
        position : axis2_placement;
    END_ENTITY; -- conic
    (*
conical_surface
    *)
    ENTITY conical_surface
        SUBTYPE OF (elementary_surface);
        radius      : length_measure;
        semi_angle  : plane_angle_measure;
        WHERE
            wr1: radius >= 0;
    END_ENTITY; -- conical_surface
    (*
connected_face_set
    *)
    ENTITY connected_face_set
        SUPERTYPE OF (ONEOF (closed_shell,open_shell))
        SUBTYPE OF (topological_representation_item);
        cfs_faces : SET [1:?] OF face;

```



```

    END_ENTITY; -- connected_face_set
  (*
context_dependent_unit
  *)
  ENTITY context_dependent_unit
    SUBTYPE OF (named_unit);
    name : label;
  END_ENTITY; -- context_dependent_unit
  (*
continuous_process_relationship
  *)
  ENTITY continuous_process_relationship
    SUBTYPE OF (sequential_method);
    WHERE
      wr1: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
        related_method);
      wr2: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
        relating_method);
      wr3: SELF.description IN ['serial','batch','serial and batch'];
  END_ENTITY; -- continuous_process_relationship
  (*
controller
  *)
  ENTITY controller
    SUBTYPE OF (action_resource);
    WHERE

  (* -- controller to specification (as controller_specification) *)

  wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    ('CAST_PARTS_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN TYPEOF
    (adr.assigned_document))))=1;

  END_ENTITY; -- controller
  (*
conversion_based_unit
  *)
  ENTITY conversion_based_unit
    SUBTYPE OF (named_unit);
    name : label;
    conversion_factor : measure_with_unit;
  END_ENTITY; -- conversion_based_unit
  (*
cooling_port
  *)
  ENTITY cooling_port
    SUBTYPE OF (die_design_feature);
    WHERE

  (* ----- male_or_female----- *)

```

```

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'male or female') AND
  ((it.description IN ['male','female'] ) ) ) ) ) =1 ) ) )>=1;

(* ----- cooling_port to item_size (as cooling_size) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='cooling port occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
AND
  (sar.description = 'cooling port size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  ) ) =1 ) ) ) ) >=1;

(* ----- cooling_port to planar_element (as port_location)----- *)

wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='port location')))=1;

END_ENTITY; -- cooling_port

```

**coordinated\_universal\_time\_offset**

```

*)
ENTITY coordinated_universal_time_offset;
  hour_offset : INTEGER;
  minute_offset : OPTIONAL INTEGER;
  sense : ahead_or_behind;
DERIVE
  actual_minute_offset : INTEGER := NVL(minute_offset,0);
WHERE
  wr1: (0 <= hour_offset) AND (hour_offset < 24);
  wr2: (0 <= actual_minute_offset) AND (actual_minute_offset <= 59);
  wr3: NOT ((hour_offset <> 0) OR (actual_minute_offset <> 0)) AND (
  sense = exact);
END_ENTITY; -- coordinated_universal_time_offset

```

**core**

```

*)
ENTITY core
  SUBTYPE OF (sand_cast_design_feature);
WHERE

(* ----- core_type----- *)

```

```

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'core type') AND
    ((it.description IN ['baked core','cover core','isocure','ram up core',
    'ring core','setup core','slab core','so2 core'] )
    )))) =1 ))))>=1;

(* ----- material_definition ----- *)

WR2: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)) AND
  (co.name IN ['ceramic','oil bonded sand','plaster',
  'resin coated sand']
  )) ) ) =1);

(* ----- core to location_element (as vent_placement)----- *)

wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='vent location')))=1;

(* ----- core to sand_mould (as core_placement) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core placement reference usage') AND
  ('CAST_PARTS_SCHEMA.SAND_MOULD' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

(* ----- core to vent ( as gas vent) ----- *)

WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'gas vent reference usage') AND
  ('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

END_ENTITY; -- core
(*
core_assembly
*)

```

## ISO 10303-223:2008 (E)

```
ENTITY core_assembly
  SUBTYPE OF (casting_method);
  WHERE

  (* ----- core_assembly to descriptive_parameter (as hard_form) -----*)

  wr1: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
      TYPEOF(pdr.representation)) AND
      (SIZEOF(QUERY (it <* pdr.representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
          IN TYPEOF(it)) AND ((it.name = 'hard form')
            ))) =1 ))))>=1;

  (* ----- core_assembly to proof_report (as soft_copy) -----*)

  wr2: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS' IN
    TYPEOF(SELf)) AND (NOT (SIZEOF(QUERY ( doc <* SELf.
      documents | (doc.kind.product_data_type = 'proof report') ))
      = 0));
END_ENTITY; -- core_assembly
```

(\*

### core\_bench

\*)

```
ENTITY core_bench
  SUBTYPE OF (core_equipment);
  WHERE

  (* ----- core_required----- *)
  wr1: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
      TYPEOF(pdr.representation)) AND
      (SIZEOF(QUERY (it <* pdr.representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
          IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'core required')) =1 ))))=1;
```

(\* ----- wax\_vent----- \*)

```
wr2: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND ((it.name = 'wax vent')
          ))) =1 ))))>=1;
```

END\_ENTITY; -- core\_bench

(\*

### core\_box\_tooling

\*)

```
ENTITY core_box_tooling
  SUBTYPE OF (casting_product_definition);
```

```

WHERE
(* core_box_tooling to production_core_box(as cores_for_pattern) *)

wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core box tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production core box reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
  IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;

END_ENTITY; -- core_box_tooling
(*
core_box_vent
*)
ENTITY core_box_vent
  SUBTYPE OF (representation);
WHERE

(* ---- material_type ---- *)

wr1: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'material type') AND
  (it.description IN ['brass','nylon',
  'polyurethane','stainless steel'] )
  )) =1;

(* ---- vent_type ---- *)

wr2: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'vent type')
  )) =1;

(* ---- core_box_vent to equipment_size (as dimensions) ---- *)

wr3: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'diameter'))
  ))=1;

wr4: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'draft'))
  ))=1;

```

```

    ))=1;

wr5: SIZEOF(QUERY ( it <* SELF.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'length'))
    ))=1;

wr6: SIZEOF(QUERY ( it <* SELF.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'width'))
    ))=1;

wr7: SIZEOF(QUERY ( it <* SELF.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND (it.name = 'height'))
    ))=1;

(* ----- core_box_vent to shape_element (as core_shape) ----- *)

wr8: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION'
    IN TYPEOF(ap.rep_2)) ))=1;

    END_ENTITY; -- core_box_vent
(*
core_conveyor_system
*)
    ENTITY core_conveyor_system
        SUBTYPE OF (core_equipment);
    WHERE
    (* ----- cores_required----- *)

wr1: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'core required')) ) = 1 ))))=1;

(* ----- cycle_time----- *)

wr2: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'cycle time')

```

```

)))) <=1 ))))>=1;

(* ----- core_conveyer_system to equipment_size (as core_box_skid)----- *)

wr3: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid height')) ) ) <=1 ))))>=1;

wr4: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid length')) ) ) <=1 ))))>=1;

wr5: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid width')) ) ) <=1 ))))>=1;

wr6: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid diameter')) ) ) <=1 ))))>=1;

wr7: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core box skid draft')) ) ) <=1 ))))>=1;

(* ----- core_conveyer_system to equipment_size (as core_plug)----- *)

wr8: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN

```

```

        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug height')) ) <=1 ))))>=1;

wr9: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug length')) ) <=1 ))))>=1;

wr10: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug width')) ) <=1 ))))>=1;

wr11: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug diameter')) ) <=1 ))))>=1;

wr12: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'core plug draft')) ) <=1 ))))>=1;

    END_ENTITY; -- core_conveyer_system
(*)
core_equipment
*)
    ENTITY core_equipment
        SUBTYPE OF (casting_equipment);
    END_ENTITY; -- core_equipment
(*)
core_machine
*)
1472

```



```

ENTITY core_machine
  SUBTYPE OF (core_equipment);
  WHERE
(* ----- core_machine to core_box_vent (as vent definition)----- *)

wr1:    SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.CORE_BOX_VENT' IN
  TYPEOF(pdr.representation))))=1;

(* ----- core_machine ejector_equipment(as ejector_definition)--- *)

wr2:    SIZEOF(QUERY ( arr <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE') |
  (('CAST_PARTS_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP' IN TYPEOF(arr)) AND
  ('CAST_PARTS_SCHEMA.EJECTOR_EQUIPMENT' IN TYPEOF(arr.relatinq_resource)
  ))) = 1;

(* --- core_machine to machine_mounting (as tool_mounting_definition)---- *)

wr3:    SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.MACHINE_MOUNTING' IN
  TYPEOF(pdr.representation)) ))=1;

```

```
END_ENTITY; -- core_machine
```

```
(*
core_master
*)
```

```
ENTITY core_master
  SUBTYPE OF (core_box_tooling);
  WHERE
```

```
wr1:    SIZEOF(QUERY ( pdr <* get_property_definition_representations(
  SELf) | (
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) )) = 1;
```

```
(* -----core box master_shrink_factor----- *)
```

```
wr2:    SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'core box master shrink factor')) )) =1 ))))=1;
```

```
END_ENTITY; -- core_master
```

```
(*
core_print
*)
```

```
ENTITY core_print
```

```
© ISO 2008 — All rights reserved
```

1473

ISO 10303-223:2008 (E)

```

SUBTYPE OF (sand_cast_design_feature);
WHERE
(* core_print to numeric_parameter(as clearance) *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'clearance')) ) =1 )))>=1;

(* core_print to drafted_surface (as draft) *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.DRAFTED_SURFACE' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='draft')))=1;

(* ----- core_print to item_size (as core_print_dimensions)----- *)

WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core print occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
AND
  (sar.description = 'core print size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 ))) >=1;

(* ----- core_print to shape_element (as feature)----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='core print occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 ))) >=1;

END_ENTITY; -- core_print
(*
core_setter
*)
ENTITY core_setter
  SUBTYPE OF (core_equipment);
  WHERE

```

```

wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF)
|
      (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
      TYPEOF(pdr.representation)) AND
      (SIZEOF(QUERY (it <* pdr.representation.items |
      (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND ((it.name = 'moulding machine')
      )))) =1 ))))>=1;
END_ENTITY; -- core_setter
(*
corrosion_service
*)
ENTITY corrosion_service
  SUBTYPE OF (customer_casting_requirement);
  WHERE
  (* ---- to descriptive_parameter(as corrosive_environment) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'corrosive environment')
  ) ) =1 )))=0));

(* ----- to material (as corrosive_material)----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'corrosive material')
  ) ) =1 )))=0));

(* ----- to numeric_parameter(as ph value) ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'ph value'))
  )) = 1 )))=0));

(* ---- to numeric_parameter(as temperature) ----- *)

wr4:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND

```

ISO 10303-223:2008 (E)

```

        (it.name = 'temperature'))
    )) = 1 )))=0));

    END_ENTITY; -- corrosion_service
(*)
curve
*)
    ENTITY curve
        SUPERTYPE OF (ONEOF (line, conic, pcurve, surface_curve))
        SUBTYPE OF (geometric_representation_item);
    END_ENTITY; -- curve
(*)
customer_casting_requirement
*)
    ENTITY customer_casting_requirement
        SUBTYPE OF (property_definition);
        WHERE
(*) ----- new_tooling_required ----- *)

wr1:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'new tooling required') AND
    (it.description IN ['customers tool shop','foundry tool shop'] )
    ))) <=1 )) )) = 0);

(*) ----- specific_comments ----- *)

wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    (it.name = 'specific comments')
    ))) >=0 )) )) = 0);

(*) --- to illustration (as illustration)----- *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
    (adr.assigned_document))))>=0;

(*) --- t to specification (as referenced_from)----- *)

wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (adr.assigned_document.kind.product_data_type=
    'customer casting requirements') ))
    >=0;

(*) --- to shape_aspect (as applies_to)----- *)

```

```

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT'
  IN TYPEOF(pdr.related_property_definition.definition)
  ))>=0);

END_ENTITY; -- customer_casting_requirement
(*)
customer_simulation
*)
ENTITY customer_simulation
  SUBTYPE OF (simulation_process);
  WHERE
    wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS' IN
      TYPEOF(SELF)) AND (SIZEOF(QUERY ( doc <* SELF\
      action_method_with_associated_documents.documents | (doc.
      kind.product_data_type = 'customer tool design report') )) =
      1);
    wr3: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS' IN
      TYPEOF(SELF)) AND (SIZEOF(QUERY ( doc <* SELF\
      action_method_with_associated_documents.documents | (doc.
      kind.product_data_type = 'customer design deficiency report') ))
      = 1);
  END_ENTITY; -- customer_simulation
(*)
cylindrical_surface
*)
ENTITY cylindrical_surface
  SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface
(*)
cylindricity_tolerance
*)
ENTITY cylindricity_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: NOT (('CAST_PARTS_SCHEMA.' +
      'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
  END_ENTITY; -- cylindricity_tolerance
(*)
data_curve
*)
ENTITY data_curve
  SUBTYPE OF (representation);
  WHERE
    (* --- interpolation method----- *)

wr1: SIZEOF(QUERY( sri <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF (sri))
  AND (sri.name='interpolation method')))=1;

```

## ISO 10303-223:2008 (E)

```
(* --- x unit----- *)

wr2: SIZEOF(QUERY( sri <* SELF.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (sri))
  AND (sri.name='x unit')))=1;

(* --- y unit----- *)

wr3: SIZEOF(QUERY( sri <* SELF.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (sri))
  AND (sri.name='y unit')))=1;

(* --- data_curve to curve_point (as data_points)----- *)

wr4: SIZEOF(QUERY( sri <* SELF.items |
  (('CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN TYPEOF (sri))
  ))>=1;

END_ENTITY; -- data_curve
(*)
data_environment
*)
ENTITY data_environment;
  name : label;
  description : text;
  elements : SET [1:?] OF property_definition_representation;
END_ENTITY; -- data_environment
(*)
date
*)
ENTITY date
  SUPERTYPE OF (calendar_date);
  year_component : year_number;
END_ENTITY; -- date
(*)
date_and_time
*)
ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY; -- date_and_time
(*)
date_and_time_assignment
*)
ENTITY date_and_time_assignment
  ABSTRACT SUPERTYPE;
  assigned_date_and_time : date_and_time;
  role : date_time_role;
END_ENTITY; -- date_and_time_assignment
(*)
```

**date\_assignment**

```

*)
ENTITY date_assignment
  ABSTRACT SUPERTYPE;
  assigned_date : date;
  role          : date_role;
END_ENTITY; -- date_assignment

```

```

(*)

```

**date\_role**

```

*)
ENTITY date_role;
  name : label;
DERIVE
  description : text := get_description_value(SELF);
WHERE
  wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_role

```

```

(*)

```

**date\_time\_role**

```

*)
ENTITY date_time_role;
  name : label;
DERIVE
  description : text := get_description_value(SELF);
WHERE
  wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
    'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_time_role

```

```

(*)

```

**dated\_effectivity**

```

*)
ENTITY dated_effectivity
  SUBTYPE OF (effectivity);
  effectivity_end_date : OPTIONAL date_time_or_event_occurrence;
  effectivity_start_date : date_time_or_event_occurrence;
END_ENTITY; -- dated_effectivity

```

```

(*)

```

**datum**

```

*)
ENTITY datum
  SUBTYPE OF (shape_aspect);
  identification : identifier;
INVERSE
  established_by_relationships : SET [1:?] OF
    shape_aspect_relationship FOR
    related_shape_aspect;
WHERE
  wr1: SIZEOF(QUERY ( x <* SELF.established_by_relationships | (
    SIZEOF(TYPEOF(x.relying_shape_aspect) * [
    'CAST_PARTS_SCHEMA.DATUM_FEATURE',

```

```

        'CAST_PARTS_SCHEMA.DATUM_TARGET']) (<> 1) )) = 0;
    END_ENTITY; -- datum
    (*
datum_feature
    *)
    ENTITY datum_feature
        SUBTYPE OF (shape_aspect);
        INVERSE
            feature_basis_relationship : shape_aspect_relationship FOR
                relating_shape_aspect;

        WHERE
            wr1: SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATING_SHAPE_ASPECT')) | (NOT ('CAST_PARTS_SCHEMA.DATUM'
                IN TYPEOF(sar.related_shape_aspect)))) )) = 0;
            wr2: SELF.product_definitional = TRUE;
    END_ENTITY; -- datum_feature
    (*
datum_reference
    *)
    ENTITY datum_reference;
        precedence : INTEGER;
        referenced_datum : datum;
        WHERE
            wr1: precedence > 0;
    END_ENTITY; -- datum_reference
    (*
datum_target
    *)
    ENTITY datum_target
        SUBTYPE OF (shape_aspect);
        target_id : identifier;
        INVERSE
            target_basis_relationship : shape_aspect_relationship FOR
                relating_shape_aspect;

        WHERE
            wr1: SIZEOF(QUERY ( sar <* bag_to_set(USEDIN(SELF,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATING_SHAPE_ASPECT')) | (NOT ('CAST_PARTS_SCHEMA.DATUM'
                IN TYPEOF(sar.related_shape_aspect)))) )) = 0;
            wr2: SELF.product_definitional = TRUE;
    END_ENTITY; -- datum_target
    (*
defect_prediction
    *)
    ENTITY defect_prediction
        SUBTYPE OF (action_method);
        WHERE
            (* defect_prediction to location_element(as location) *)

            wr1: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |

```



```

(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
 (NOT (SIZEOF(QUERY(eds <* ama.items |
 ('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
 (NOT (SIZEOF(QUERY(pd <* USEDIN(eds,
 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 (SIZEOF(QUERY(mri <* pd.used_representation.items |
 ('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN TYPEOF(mri))))=1)
 ))=1))))=0))))=0));

(* defect_prediction to shape_aspect(as defect_shape) *)

wr2: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
 'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
 (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
 (SIZEOF(QUERY(eds <* ama.items |
 ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
 )) =1 )) ))=0));

```

```
END_ENTITY; -- defect_prediction
```

```
(*
defining_action_method_relationship
*)
```

```
ENTITY defining_action_method_relationship
  SUBTYPE OF (action_method_relationship);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
      related_method);
  END_ENTITY; -- defining_action_method_relationship
```

```
(*
definitional_representation
*)
```

```
ENTITY definitional_representation
  SUBTYPE OF (representation);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.PARAMETRIC_REPRESENTATION_CONTEXT' IN
      TYPEOF(SELF\representation.context_of_items);
  END_ENTITY; -- definitional_representation
```

```
(*
degenerate_toroidal_surface
*)
```

```
ENTITY degenerate_toroidal_surface
  SUBTYPE OF (toroidal_surface);
  select_outer : BOOLEAN;
  WHERE
    wr1: major_radius < minor_radius;
  END_ENTITY; -- degenerate_toroidal_surface
```

```
(*
derived_shape_aspect
*)
```

```
ENTITY derived_shape_aspect
  SUPERTYPE OF (ONEOF (apex,centre_of_symmetry,geometric_alignment,
    geometric_intersection,parallel_offset,perpendicular_to,extension,
```

```

        tangent))
    SUBTYPE OF (shape_aspect);
    INVERSE
        deriving_relationships : SET [1:?] OF
            shape_aspect_deriving_relationship FOR
            relating_shape_aspect;
    END_ENTITY; -- derived_shape_aspect
(*)
derived_unit
*)
    ENTITY derived_unit;
        elements : SET [1:?] OF derived_unit_element;
    DERIVE
        name : label := get_name_value(SELF);
    WHERE
        wr1: (SIZEOF(elements) > 1) OR ((SIZEOF(elements) = 1) AND (elements
            [1].exponent <> 1));
        wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
    END_ENTITY; -- derived_unit
(*)
derived_unit_element
*)
    ENTITY derived_unit_element;
        unit      : named_unit;
        exponent  : REAL;
    END_ENTITY; -- derived_unit_element
(*)
description_attribute
*)
    ENTITY description_attribute;
        attribute_value : text;
        described_item  : description_attribute_select;
    END_ENTITY; -- description_attribute
(*)
descriptive_representation_item
*)
    ENTITY descriptive_representation_item
        SUBTYPE OF (representation_item);
        description : text;
    END_ENTITY; -- descriptive_representation_item
(*)
design_part
*)
    ENTITY design_part
        SUBTYPE OF (product_definition);
    WHERE
        wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
            ( 'CAST_PARTS_SCHEMA.INSTANCED_FEATURE' IN TYPEOF (sa)
            )))>=1;
    END_ENTITY; -- design_part

```

(\*

## design\_reference

\*)

```

ENTITY design_reference
  SUBTYPE OF (document);
  WHERE
    wr1: SIZEOF(QUERY ( duc <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
      (duc.subject_element = 'drawing_revision_level')) = 1;

    wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
      ('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))) |
      (NOT(SIZEOF(QUERY ( d <* adr.items |
      ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d))
      )) >= 1)))) = 0;

    wr3: SIZEOF(QUERY(dpa <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT') |
      (SIZEOF(['CAST_PARTS_SCHEMA.DESIGN_PART',
      'CAST_PARTS_SCHEMA.CAST_PART',
      'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
      'CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION']
      * TYPEOF(dpa.related_product))=1)
      ))=1;

  END_ENTITY; -- design_reference

```

(\*

## die\_cast\_master

\*)

```

ENTITY die_cast_master
  SUBTYPE OF (die_cast_tooling);
  WHERE

  (* -----master_shrink_factor----- *)

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY (it <* pdr.used_representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
      * TYPEOF(it)) = 2) AND
      (it.name = 'master shrink factor')) = 1 ))))=1;

  (* ***** to production_die_cast_mould (as female_component) ***** *)

    wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
      ((sa.description='die cast tooling occurrence') AND
      (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |

```

```

        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'production die cast mould female reference usage')
AND
        ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
        IN TYPEOF(sar.relater_shape_aspect))
        )) =1 )))) >=1;

```

(\* \*\* die\_cast\_master to production\_die\_cast\_mould (as male\_component) \*)

```

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='die cast tooling occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'production die cast mould male reference usage')
AND
        ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
        IN TYPEOF(sar.relater_shape_aspect))
        )) =1 )))) >=1;

```

(\* \*\* to machining\_allowance (as master\_allowance\_on\_master\_tooling) \* \*)

```

wr5: (SIZEOF(QUERY (co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))=1);
END_ENTITY; -- die_cast_master

```

(\*  
**die\_cast\_tooling**

\*)  
 ENTITY die\_cast\_tooling  
 SUBTYPE OF (casting\_product\_definition);  
 WHERE

(\* ----- die\_cast\_tooling to material (as material\_definition) \*)

```

WR1: SIZEOF(QUERY (pdr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.'+
        'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
        (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
        (pdr.name='material definition') AND
        (SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
        ))=1))))=1;

```

(\* \*\*\*\*\* to production\_die\_cast\_mould (as die\_equipment) \*\*\*\*\* \*)

```

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='die cast tooling occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'production die cast mould reference usage') AND

```

```

('CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD'
IN TYPEOF(sar.relatng_shape_aspect))
)) >=1 )))) >=1;

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr3: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELf) |
( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
IN TYPEOF ( pdr.used_representation ) ))) =1) OR
(SIZEOF(QUERY(ada <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
(ada.role.name='tooling_location')))=1);

END_ENTITY; -- die_cast_tooling
(*
die_clamping
*)
ENTITY die_clamping
SUBTYPE OF (die_design_feature);
WHERE

(* *** die_clamping to orientation (as die_clamping_positions) ***** *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
(it.name = 'clamping_position')))) =1 ))))=1;

END_ENTITY; -- die_clamping
(*
die_core
*)
ENTITY die_core
SUBTYPE OF (die_design_feature);
WHERE

(* ----- core_type----- *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND ((it.name = 'core type') AND
((it.description IN ['fixed core','hydraulic slide',
'mechanical slide','sand core',
'salt core','ceramic'] )) )) =1 )) ))>=1;

(* ***** metal_type ***** *)

```

ISO 10303-223:2008 (E)

```

wr2: (SIZEOF(QUERY (co <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

(* die_core to drafted_surface (as draft) *)

wr3: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ))) =1;

(* ----- die_core to item_size (as guide_pin) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP')
  IN TYPEOF(sar)) AND
  (sar.description = 'guide pin size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

(* ----- die_core to parting_surface (as parting) ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  ('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF(pdr.used_representation)) )) = 1;

(* ----- die_core to shape_aspect (as outside_shape_of_die_mould)----- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die core occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'die core area shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 )))) >=1;

  END_ENTITY; -- die_core
  (*
die_design_feature
  *)
  ENTITY die_design_feature
    SUBTYPE OF (casting_feature_definition);
    WHERE

WR1: SIZEOF(QUERY( pdr <* get_property_definition_representations (SELF) |
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF ( pdr.used_representation ) ) ) =1;

```

```

(* ***** die_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 ))))=1;

  END_ENTITY; -- die_design_feature
(*
die_master
*)
  ENTITY die_master
    SUBTYPE OF (die_mould_tooling);
    WHERE

(* -----master_shrink_factor----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'master shrink factor')) ) =1 ))))=1;

(* * die_master to production_die_cast_mould (as female_component) ** *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die master occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould female reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 )))) >=1;

(* * die_master to production_die_cast_mould (as male_component) ***** *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die master occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould male reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 )))) >=1;

(* * to machining_allowance (as machining_allowance_on_master_tooling) *)

```

ISO 10303-223:2008 (E)

```
wr5: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION' ) |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))>=0);
END_ENTITY; -- die_master
(*)
die_mould_tooling
*)
ENTITY die_mould_tooling
  SUBTYPE OF (casting_product_definition);
  WHERE

  (* ----- die_mould_tooling to material (as material_definition)  *)

WR1: SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION' ) |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='material_definition') AND
  (SIZEOF(QUERY (mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS' ) |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
  ))=1))))=1;

  (* ** die_mould_tooling to production_die_mould (as die_equipment)  *** *)

Wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='die mould tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT' ) |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production die mould reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD'
  IN TYPEOF(sar.relying_shape_aspect))
  )) >=2 )))) >=1;

  (* ***** to in_facility_location (as tooling_location)  ***** *)
  (* ***** to person_and_organization (as tooling_location)  ***** *)

Wr3: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ))) >=1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS' ) |
  (ada.role.name='tooling_location'))>=1);
END_ENTITY; -- die_mould_tooling
(*)
die_mould_vent
*)
ENTITY die_mould_vent
  SUBTYPE OF (die_design_feature);
  WHERE
    wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
```



```

        ((sa.description='die mould vent occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'ejector pin reference usage') AND
        ('CAST_PARTS_SCHEMA.EJECTOR_PIN' IN TYPEOF(sar.relatng_shape_aspect))
        )) >=1 )))) >=1;

    END_ENTITY; -- die_mould_vent
(*)
dimension_related_tolerance_zone_element
*)
    ENTITY dimension_related_tolerance_zone_element;
        related_dimension : dimensional_location;
        related_element    : tolerance_zone_definition;
    END_ENTITY; -- dimension_related_tolerance_zone_element
(*)
dimensional_characteristic_representation
*)
    ENTITY dimensional_characteristic_representation;
        dimension          : dimensional_characteristic;
        representation      : shape_dimension_representation;
    END_ENTITY; -- dimensional_characteristic_representation
(*)
dimensional_exponents
*)
    ENTITY dimensional_exponents;
        length_exponent           : REAL;
        mass_exponent              : REAL;
        time_exponent              : REAL;
        electric_current_exponent  : REAL;
        thermodynamic_temperature_exponent : REAL;
        amount_of_substance_exponent : REAL;
        luminous_intensity_exponent : REAL;
    END_ENTITY; -- dimensional_exponents
(*)
dimensional_location
*)
    ENTITY dimensional_location
        SUPERTYPE OF (ONEOF (angular_location,dimensional_location_with_path))
        SUBTYPE OF (shape_aspect_relationship);
    END_ENTITY; -- dimensional_location
(*)
dimensional_location_with_path
*)
    ENTITY dimensional_location_with_path
        SUBTYPE OF (dimensional_location);
        path : shape_aspect;
    END_ENTITY; -- dimensional_location_with_path
(*)
dimensional_measurement_inspection

```

## ISO 10303-223:2008 (E)

```
*)
  ENTITY dimensional_measurement_inspection
    SUBTYPE OF (inspection_activity);
    WHERE
(* -- to external_schema_definition (as iso10303_219_data) ----- *)

wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (SIZEOF(QUERY ( edi <* sar.items |
  ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA'
  IN TYPEOF(edi)) )) <= 1))))>=0;

  END_ENTITY; -- dimensional_measurement_inspection
(*
```

### **dimensional\_size**

```
*)
  ENTITY dimensional_size
    SUPERTYPE OF (ONEOF (angular_size,dimensional_size_with_path));
    applies_to : shape_aspect;
    name       : label;
    WHERE
      wr1: applies_to.product_definitional = TRUE;
  END_ENTITY; -- dimensional_size
(*
```

### **dimensional\_size\_with\_path**

```
*)
  ENTITY dimensional_size_with_path
    SUBTYPE OF (dimensional_size);
    path : shape_aspect;
  END_ENTITY; -- dimensional_size_with_path
(*
```

### **directed\_action**

```
*)
  ENTITY directed_action
    SUBTYPE OF (executed_action);
    directive : action_directive;
  END_ENTITY; -- directed_action
(*
```

### **directed\_dimensional\_location**

```
*)
  ENTITY directed_dimensional_location
    SUBTYPE OF (dimensional_location);
  END_ENTITY; -- directed_dimensional_location
(*
```

### **direction**

```
*)
  ENTITY direction
    SUBTYPE OF (geometric_representation_item);
    direction_ratios : LIST [2:3] OF REAL;
```

```

WHERE
    wr1: SIZEOF(QUERY ( tmp <* direction_ratios | (tmp <> 0) )) > 0;
END_ENTITY; -- direction
(*)
direction_shape_representation
*)
ENTITY direction_shape_representation
    SUBTYPE OF (shape_representation);
WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
        'CAST_PARTS_SCHEMA.DIRECTION' IN TYPEOF(it))) )) = 0;
END_ENTITY; -- direction_shape_representation
(*)
document
*)
ENTITY document;
    id          : identifier;
    name        : label;
    description  : OPTIONAL text;
    kind        : document_type;
INVERSE
    representation_types : SET [0:?] OF document_representation_type FOR
        represented_document;
END_ENTITY; -- document
(*)
document_file
*)
ENTITY document_file
    SUBTYPE OF (document, characterized_object);
WHERE
    wr1 : SIZEOF(QUERY ( drt <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT') |
        (drt.name IN ['physical','digital']) )) = 1;

    wr2 : SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS') |
        (('CAST_PARTS_SCHEMA.IDENTIFICATION_ASSIGNMENT'
        IN TYPEOF(adr)) AND (adr.role.name = 'version')) )) <= 1;

    wr3: SIZEOF(QUERY ( aeia <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
        ('CAST_PARTS_SCHEMA.EXTERNAL_IDENTIFICATION_ASSIGNMENT'
        IN TYPEOF(aeia)) )) >= 0;

    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (pd.name='document property') AND (SIZEOF(QUERY ( pdr <*
        USEDIN(pd,

```

```
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.DOCUMENT_FILE_PROPERTIES'
IN TYPEOF(pdr.used_representation)) AND
(pdr.used_representation.name = 'document format')))) <=1))) <=1;
```

```
END_ENTITY; -- document_file
```

```
(*
```

### **document\_file\_properties**

```
*)
```

```
ENTITY document_file_properties
```

```
  SUBTYPE OF (representation);
```

```
  WHERE
```

```
wr1: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='country code') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr2: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='detail level') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr3: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='geometry type') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr4: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='language code') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr5: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating interface') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr6: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating operating system') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr7: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='creating system') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```
wr8: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='data format') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) )) <=1;
```

```

wr9: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='format character code') AND
  ( 'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF (r))) ) ) <=1;

wr10: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='file size') AND
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(r) = 2)))) <=1;

wr11: SIZEOF(QUERY(r <* SELF.items |
  ((r.name='page count') AND
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
  TYPEOF(r) = 2)))) <=1;

```

```
END_ENTITY; -- document_file_properties
```

```
(*
```

### **document\_product\_association**

```
*)
```

```

ENTITY document_product_association;
  name          : label;
  description    : OPTIONAL text;
  relating_document : document;
  related_product  : product_or_formation_or_definition;
END_ENTITY; -- document_product_association

```

```
(*
```

### **document\_reference**

```
*)
```

```

ENTITY document_reference
  ABSTRACT SUPERTYPE;
  assigned_document : document;
  source            : label;
  DERIVE
  role : object_role := get_role(SELF);
  WHERE
  wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
    'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- document_reference

```

```
(*
```

### **document\_relationship**

```
*)
```

```

ENTITY document_relationship;
  name          : label;
  description    : OPTIONAL text;
  relating_document : document;
  related_document : document;
END_ENTITY; -- document_relationship

```

```
(*
```

### **document\_representation\_type**

```
*)
```

© ISO 2008 — All rights reserved

```

ENTITY document_representation_type;
    name          : label;
    represented_document : document;
END_ENTITY; -- document_representation_type
(*)
document_type
*)
ENTITY document_type;
    product_data_type : label;
END_ENTITY; -- document_type
(*)
document_usage_constraint
*)
ENTITY document_usage_constraint;
    source          : document;
    subject_element : label;
    subject_element_value : text;
END_ENTITY; -- document_usage_constraint
(*)
document_usage_constraint_assignment
*)
ENTITY document_usage_constraint_assignment
    ABSTRACT SUPERTYPE;
    assigned_document_usage : document_usage_constraint;
    role                    : document_usage_role;
END_ENTITY; -- document_usage_constraint_assignment
(*)
document_usage_role
*)
ENTITY document_usage_role;
    name          : label;
    description   : OPTIONAL text;
END_ENTITY; -- document_usage_role
(*)
drafted_surface
*)
ENTITY drafted_surface
    SUBTYPE OF (shape_representation);
    WHERE

wr1: (SIZEOF(QUERY( it <* SELF.items |
    (SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'draft'))=1);

wr2:(SIZEOF(QUERY (rep <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
    (SIZEOF(['CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' ,
    'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION']
    * TYPEOF(rep.rep_2))=1)))=1);

```

```

    END_ENTITY; -- drafted_surface
  (*)
dryer
  *)
    ENTITY dryer
      SUBTYPE OF (core_equipment);
      WHERE
        wr1:    SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF)
|
          (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
            TYPEOF(pdr.representation)) AND
            (SIZEOF(QUERY (it <* pdr.representation.items |
              (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND ((it.name = 'material type')
                  ))) =1 ))))>=1;
        END_ENTITY; -- dryer
  (*)
edge
  *)
    ENTITY edge
      SUPERTYPE OF (ONEOF (edge_curve, oriented_edge))
      SUBTYPE OF (topological_representation_item);
      edge_start : vertex;
      edge_end   : vertex;
    END_ENTITY; -- edge
  (*)
edge_curve
  *)
    ENTITY edge_curve
      SUBTYPE OF (edge, geometric_representation_item);
      edge_geometry : curve;
      same_sense    : BOOLEAN;
    END_ENTITY; -- edge_curve
  (*)
edge_loop
  *)
    ENTITY edge_loop
      SUBTYPE OF (loop, path);
      DERIVE
        ne : INTEGER := SIZEOF(SELF\path.edge_list);
      WHERE
        wr1: SELF\path.edge_list[1].edge_start :=: SELF\path.edge_list[ne].
          edge_end;
    END_ENTITY; -- edge_loop
  (*)
edge_round
  *)
    ENTITY edge_round
      SUBTYPE OF (transition_feature);
      WHERE

```

```

wr1: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
    SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
wr2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
    SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | ((NOT (SIZEOF(impl_rep.
        used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
        used_representation.items) <= 3)) )) = 0)) )) = 0);
wr3: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) )) = 0);
wr4: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'first offset')) )) <= 1)) )) = 0)) )) =
    0);
wr5: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))

```



```

= 2) AND (it.name = 'second offset')) )) <= 1)) )) = 0)) ))
= 0);
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'edge round face')) )) <= 1)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'first face shape')) )) <= 1)) )) = 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'second face shape')) )) <= 1)) )) = 0;
END_ENTITY; -- edge_round
(*)
edge_shape_representation
*)
ENTITY edge_shape_representation
SUBTYPE OF (shape_representation);
END_ENTITY; -- edge_shape_representation
(*)
effectivity
*)
ENTITY effectivity
SUPERTYPE OF (ONEOF (dated_effectivity, lot_effectivity));
id : identifier;
DERIVE
name : label := get_name_value(SELf);
description : text := get_description_value(SELf);
WHERE
wr1: SIZEOF(USEDIN(SELf, 'CAST_PARTS_SCHEMA.' +
'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
wr2: SIZEOF(USEDIN(SELf, 'CAST_PARTS_SCHEMA.' +
'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- effectivity
(*)
effectivity_assignment
*)
ENTITY effectivity_assignment
ABSTRACT SUPERTYPE;
assigned_effectivity : effectivity;
DERIVE

```

```

    role : object_role := get_role(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- effectivity_assignment

(*)
ejector_box
*)
ENTITY ejector_box
    SUBTYPE OF (ejector_design_feature);
WHERE

    (* ***** ejector_box to ejector_plate (as plate_definition) ***** *)

wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='ejector box occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'ejector plate reference usage') AND
    ('CAST_PARTS_SCHEMA.EJECTOR_PLATE'
    IN TYPEOF(sar.relatng_shape_aspect))
    )) >=1 )))) >=1;

    (* ----- ejector_box to item_size (as box_size)----- *)

WR2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='ejector box occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'ejector box size usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relatng_shape_aspect))
    )) =1 )))) >=1;

    (* ***** ejector_box to *)
    (* ** production_core_box (as oriented_to_movable_die) ***** *)
    (* ** production_die_mould (as oriented_to_movable_die) ***** *)
    (* ** production_die_cast_mould (as oriented_to_movable_die) ***** *)
    (* ** production_investment_cast_mould (as oriented_to_movable_die) * *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='ejector box occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'ejector box die reference usage') AND
    ( SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX',
    'CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD',
    'CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD',
    'CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD']

```

```

        * TYPEOF(sar.relatng_shape_aspect)) =1))) =1 ))))>=1;

    END_ENTITY; -- ejector_box
  (*
ejector_design_feature
  *)
    ENTITY ejector_design_feature
      SUBTYPE OF (casting_feature_definition);
    WHERE

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation) ) ) = 1;

    (* ***** ejector_design_feature to orientation (as placement) ***** *)

    WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
          (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
          (it.name = 'orientation')))) =1 ))))=1;
    END_ENTITY; -- ejector_design_feature
  (*
ejector_equipment
  *)
    ENTITY ejector_equipment

      SUBTYPE OF (casting_equipment);
    WHERE
      wr1:(NOT (SIZEOF(QUERY( am <* SELF.usage |
        (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
        (NOT(SIZEOF(QUERY ( ama <* USEDIN(am,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
        IN TYPEOF(ama)) AND
        (SIZEOF(QUERY ( eds <* ama.items |
        ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM' IN TYPEOF(eds))
        ))=1))))=0))))=0));

    END_ENTITY; -- ejector_equipment
  (*
ejector_pin
  *)
    ENTITY ejector_pin
      SUBTYPE OF (ejector_design_feature);
    WHERE
    (* ----- ejector_pins to item_size (as pin_dimensions) ----- *)

    WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
      ((sa.description='ejector pin occurrence') AND

```

```

        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'ejector pin dimension usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
        IN TYPEOF(sar.relater_shape_aspect))
        )) =1 )))) >=1;

(* ----- ejector_pins to machining_feature (as thread) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='ejector pin occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'thread reference usage') AND
        ('CAST_PARTS_SCHEMA.THREAD' IN TYPEOF(sar.relater_shape_aspect))
        )) =1 )))) >=1;

(* ----- ejector_pins to pin_tips (as tip_definition) ----- *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='ejector pin occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'pin tip reference usage') AND
        ('CAST_PARTS_SCHEMA.PIN_TIP' IN TYPEOF(sar.relater_shape_aspect))
        )) =1 )))) >=1;

        END_ENTITY; -- ejector_pin
(*
ejector_plate
*)
        ENTITY ejector_plate
        SUBTYPE OF (ejector_design_feature);
        WHERE
(* ***** material_type ***** *)

wr1: (SIZEOF(QUERY (co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))=1);

(* ----- ejector_plate to item_size (as pin_dimensions) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='ejector plate occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'ejector plate dimension usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
        IN TYPEOF(sar.relater_shape_aspect))

```

```

    )) =1 )))) >=1;

(* ejector_plate to numeric_parameter (as plate_height) *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'plate height')) ) =1 ))))=1;

(* --- ejector_plate to ejector_pin (as ejector_pin_definition) ----- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector pin reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PIN' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;

(* ----- ejector_plate to stop_pin (as stop_pin_definition) ----- *)

wr5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'stop pin reference usage') AND
  ('CAST_PARTS_SCHEMA.STOP_PIN' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* to reset_or_return_pins (as reset_or_return_pin_definition) *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='ejector plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'reset or push pin reference usage') AND
  ('CAST_PARTS_SCHEMA.RESET_OR_PUSH_PIN'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;

  END_ENTITY; -- ejector_plate
(*
ejector_system
*)
  ENTITY ejector_system
    SUBTYPE OF (ejector_design_feature);

```

## ISO 10303-223:2008 (E)

```
WHERE
(* ----- system_operated_by----- *)

wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'system operated by') AND
  (it.description IN ['hydraulics','spring loaded',
  'simple mechanical'] )
  )))) =1 ))))>=1;

(* ----- ejector_system to ejector_plates (as plate_definition) ----- *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ejector system occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector system reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_PLATE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

END_ENTITY; -- ejector_system
(*
elementary_surface
*)
ENTITY elementary_surface
  SUPERTYPE OF (ONEOF (plane,cylindrical_surface,conical_surface,
  spherical_surface,toroidal_surface))
  SUBTYPE OF (surface);
  position : axis2_placement_3d;
END_ENTITY; -- elementary_surface
(*
elevated_temperature_service
*)
ENTITY elevated_temperature_service
  SUBTYPE OF (customer_casting_requirement);
  WHERE
(* ----- to descriptive_parameter(as service_descriptions) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'service description')
  ) ) ) =1 ))))=0));

(* ----- to numeric_parameter(as maximum_temperature) ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
1502
```

© ISO 2008—All rights reserved

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'maximum temperature'))
)) = 1 ))=0));

(* ----- to numeric_parameter(as minimum_cyclic_temperature) ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'minimum cyclic temperature'))
)) = 1 ))=0));

(* ----- to numeric_parameter(as maximum_cyclic_temperature) ----- *)

wr4:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
* TYPEOF(it)) = 2) AND
(it.name = 'maximum cyclic temperature'))
)) = 1 ))=0));

END_ENTITY; -- elevated_temperature_service

```

```

(*

```

## ellipse

```

*)
ENTITY ellipse
  SUBTYPE OF (conic);
  semi_axis_1 : positive_length_measure;
  semi_axis_2 : positive_length_measure;
END_ENTITY; -- ellipse

```

```

(*

```

## casting\_equipment\_process

```

*)
ENTITY casting_equipment_process
  SUBTYPE OF (machining_process);
  WHERE
    wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') | (
      'CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(sar)) )) = 1;
END_ENTITY; -- casting_equipment_process

```

```

(*

```

## equipment\_setting\_record

```

*)

```

## ISO 10303-223:2008 (E)

```
ENTITY equipment_setting_record
  SUBTYPE OF (activity_result_record);
  WHERE
    (* ----- to property_parameter (as actual_machine_setting) -----*)
```

```
wr1: (SIZEOF(QUERY( it <* SELF.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM']
  * TYPEOF(it)) = 1) AND
  (it.name = 'actual_parameter')))=1);
```

```
(* ----- to equipment_usage (as recorded_usage) -----*)
```

```
wr2: (SIZEOF(QUERY( rpr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.REPRESENTATION') |
  ('CAST_PARTS_SCHEMA.EQUIPMENT_USAGE'
  IN TYPEOF (rpr.property.resource))))=1);
```

```
END_ENTITY; -- equipment_setting_record
```

```
(*
```

### casting\_equipment\_setup

```
*)
```

```
ENTITY casting_equipment_setup
  SUBTYPE OF (machine_setup);
  WHERE
```

```
wr1: (NOT(SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
  (('CAST_PARTS_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
  IN TYPEOF(sar)) AND
  (SIZEOF(QUERY( equ <* sar.resources |
  ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(equ)))
  =1 ))))=0));
```

```
END_ENTITY; -- casting_equipment_setup
```

```
(*
```

### equipment\_usage

```
*)
```

```
ENTITY equipment_usage
  SUBTYPE OF (requirement_for_action_resource);
  WHERE
```

```
(* ----- to equipment (as equipment_usage_context) -----*)
```

```
wr1: (SIZEOF(QUERY( equ <* SELF.resources |
  ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(equ))))=1);
```

```
(* ----- to property_parameter (as equipment_parameter) -----*)
```

```
wr2: SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
```



```

        'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
        * TYPEOF(it) = 1))) =1 ))) >=1;

    END_ENTITY; -- equipment_usage
  (*)
executed_action
  *)
    ENTITY executed_action
      SUBTYPE OF (action);
    END_ENTITY; -- executed_action
  (*)
extension
  *)
    ENTITY extension
      SUBTYPE OF (derived_shape_aspect);
      WHERE
        wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
    END_ENTITY; -- extension
  (*)
external_identification_assignment
  *)
    ENTITY external_identification_assignment
      ABSTRACT SUPERTYPE
      SUBTYPE OF (identification_assignment);
      source : external_source;
    END_ENTITY; -- external_identification_assignment
  (*)
external_source
  *)
    ENTITY external_source;
      source_id : source_item;
      DERIVE
        description : text := get_description_value(SELF);
      WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
          'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    END_ENTITY; -- external_source
  (*)
externally_defined_class
  *)
    ENTITY externally_defined_class
      SUBTYPE OF (externally_defined_item, class);
      WHERE
        wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

        wr2: SELF.source.name = 'ISO 13584 library';

  (* class_BSU to supplier_BSU *)

        wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF.source,
          'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |

```

```

        (aoa.role.name = 'library supplier')) = 1;

(* class_BSU.version *)

wr4: SIZEOF(QUERY ( aoa <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
    (('CAST_PARTS_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
    IN TYPEOF(aoa)) AND (aoa.role.name = 'class version')))) = 1;

```

END\_ENTITY; -- externally\_defined\_class

(\*

**externally\_defined\_dimension\_definition**

\*)

```

ENTITY externally_defined_dimension_definition
  SUBTYPE OF (externally_defined_item, dimensional_size);
  WHERE
  wr1: SELF.source.description =
    'externally defined dimension specification';

wr2: SIZEOF(QUERY ( adr <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
    (adr.assigned_document.description =
    'externally defined size dimension specification')
    ) ) <= 1;

```

END\_ENTITY; -- externally\_defined\_dimension\_definition

(\*

**externally\_defined\_feature\_definition**

\*)

```

ENTITY externally_defined_feature_definition
  SUBTYPE OF (feature_definition, externally_defined_item);
  WHERE
  wr1 : (((((SELF\characterized_object.description = 'thread') AND (
    SELF\externally_defined_item.item_id = 'external thread'))
    AND (SELF\externally_defined_item.source.source_id =
    'external feature specification')) OR (((SELF\
    characterized_object.description = 'gear') AND (SELF\
    externally_defined_item.item_id = 'external gear')) AND (
    SELF\externally_defined_item.source.source_id =
    'external feature specification')) OR (((SELF\
    characterized_object.description = 'marking') AND (SELF\
    externally_defined_item.item_id = 'external marking')) AND
    (SELF\externally_defined_item.source.source_id =
    'external feature specification')) OR (((SELF\
    characterized_object.description = 'knurl') AND (SELF\
    externally_defined_item.item_id = 'external knurl')) AND (
    SELF\externally_defined_item.source.source_id =
    'external feature specification')));
  wr2 : (NOT (SELF\characterized_object.description = 'thread')) OR (
    SIZEOF(QUERY ( pd <* USEDIN(SELf,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((

```

```

'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND ((5 <= SIZEOF(pdr.
used_representation.items)) AND (SIZEOF(pdr.
used_representation.items) <= 10))) ) = 1) ) = 1);
wr3 : (NOT (SELF\characterized_object.description = 'marking')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(pdr.used_representation.
items) = 2)) )) = 1) )) = 1);
wr4 : (NOT (SELF\characterized_object.description = 'knurl')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(pdr.used_representation.
items) = 1)) )) = 1) )) = 1);
wr5 : (NOT (SELF\characterized_object.description IN ['knurl',
'thread'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(
SELF, 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'partial area occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'applied area usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
relating_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
wr6 : (NOT (SELF\characterized_object.description = 'marking')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) )) = 0)) ))
= 0);
wr7 : (NOT (SELF\characterized_object.description = 'thread')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT

```

```

        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'removal direction')) AND ((it.
        description = 'internal') OR (it.description = 'external')))) ))
        = 1)) )) = 0)) )) = 0);
wr8 : (NOT (SELF\characterized_object.description = 'thread')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) = 0)) ))
        = 0);
wr9 : (NOT (SELF\characterized_object.description = 'thread')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'hand')) )) = 1)) )) = 0)) )) =
        0);
wr10: (NOT (SELF\characterized_object.description = 'thread')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
        = 0);
wr11: (NOT (SELF\characterized_object.description = 'thread')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT

```

```

(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) )) =
0);
wr12: (NOT (SELF\characterized_object.description = 'thread')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'major diameter')) )) <= 1)) )) = 0)) ))
= 0);
wr13: (NOT (SELF\characterized_object.description = 'thread')) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'number of threads')) )) = 1)) )) = 0)) ))
= 0);
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) )) = 0)) ))
= 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'nominal size')) ) ) <= 1)) ) = 0)) )
= 0;
wr16: (NOT (SELF\characterized_object.description IN ['knurl','gear',
'thread'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(
SELF,'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'applied_shape') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
relating_shape_aspect)) ) ) = 1) ) <= 1)) ) = 0);
wr17: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (10 <= SIZEOF(pdr.
used_representation.items)) AND (SIZEOF(pdr.
used_representation.items) >= 11)) ) ) = 1) ) = 1);
wr18: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'number of teeth')) ) ) = 1)) ) = 0)) )
= 0);
wr19: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'reference pressure angle')) ))
= 1)) )) = 0)) )) = 0);
wr20: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'rake shift factor')) )) = 1)) )) = 0)) ))
= 0);
wr21: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'nominal tool depth')) )) = 1)) )) = 0)) ))
= 0);
wr22: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'face width')) )) = 1)) )) = 0)) )) = 0);
wr23: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'tip diameter')) ) = 1)) ) = 0)) ) =
0);
wr24: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'module or diametral pitch'))
AND ((it.description = 'module') OR (it.description =
'diameter pitch')))) ) = 1)) ) = 0)) ) = 0);
wr25: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'normal attribute')) ) = 1)) ) = 0)) ) =
0);
wr26: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'internal or external gear'))
AND ((it.description = 'internal') OR (it.description =
'external')))) ) = 1)) ) = 0)) ) = 0);
wr27: (NOT (SELF\characterized_object.description IN ['gear'])) OR (

```



```

        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'root fillet radius')) ))
        <= 1)) )) = 0)) )) = 0);
    END_ENTITY; -- externally_defined_feature_definition
(*
externally_defined_general_property
*)
    ENTITY externally_defined_general_property
        SUBTYPE OF (general_property, externally_defined_item);
        WHERE
        wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

        wr2: SELF.source.name = 'ISO 13584 library';

        (* property_BSU.version *)

        wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
        (('CAST_PARTS_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
        IN TYPEOF(aoa)) AND
        (aoa.role.name = 'property version')))) = 1;

        wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM') |
        ((ap.name = 'name scope') AND
        ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_CLASS' IN
        TYPEOF(ap.related_item)))))) >= 1;

    END_ENTITY; -- externally_defined_general_property
(*
externally_defined_item
*)
    ENTITY externally_defined_item;
        item_id : source_item;
        source : external_source;
    END_ENTITY; -- externally_defined_item
(*
externally_defined_item_relationship
*)
    ENTITY externally_defined_item_relationship;
        name : label;
        description : OPTIONAL text;

```

```

        relating_item : externally_defined_item;
        related_item  : externally_defined_item;
    END_ENTITY; -- externally_defined_item_relationship
    (*
externally_defined_representation_with_parameters
    *)
    ENTITY externally_defined_representation_with_parameters
        SUBTYPE OF (representation);
    WHERE
        wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') |
            (('CAST_PARTS_SCHEMA.APPLIED_LIBRARY_ASSIGNMENT' IN TYPEOF(adr)) AND
            (adr.role.name='definitional class membership')))= 1;

        wr2: SIZEOF(QUERY ( adr <* SELF.items |
            ('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(adr)) )) = 1;

        wr3: SIZEOF(QUERY ( adr <* SELF.items |
            (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(adr)) AND
            ('CAST_PARTS_SCHEMA.CARTESIAN_POINT'
            IN TYPEOF(adr.location))) )) = 1;

    END_ENTITY; -- externally_defined_representation_with_parameters
    (*

```

### **externally\_defined\_schema**

```

    *)
    ENTITY externally_defined_schema
        SUBTYPE OF (externally_defined_item);
    WHERE
        wr1: 'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);

        wr2: SELF.source.name = 'ISO 10303 part';

        wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
            (adr.assigned_document.description = 'externally defined schema') ))
            <= 1;

        wr4: SELF.item_id IN ['externally defined schema','workingstep',
            'nc function'];

    END_ENTITY; -- externally_defined_schema
    (*

```

### **fabrication**

```

    *)
    ENTITY fabrication
        SUBTYPE OF (non_casting_activity);
    END_ENTITY; -- fabrication
    (*
    face
    *)
    ENTITY face

```

```

SUPERTYPE OF (ONEOF (face_surface,oriented_face))
SUBTYPE OF (topological_representation_item);
  bounds : SET [1:?] OF face_bound;
WHERE
  wr1: NOT mixed_loop_type_set(list_to_set(list_face_loops(SELF)));
  wr2: SIZEOF(QUERY ( temp <* bounds | (
    'CAST_PARTS_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(temp)) )) <=
    1;
END_ENTITY; -- face
(*)
face_bound
*)
ENTITY face_bound
  SUBTYPE OF (topological_representation_item);
  bound : loop;
  orientation : BOOLEAN;
END_ENTITY; -- face_bound
(*)
face_outer_bound
*)
ENTITY face_outer_bound
  SUBTYPE OF (face_bound);
END_ENTITY; -- face_outer_bound
(*)
face_shape_representation
*)
ENTITY face_shape_representation
  SUBTYPE OF (shape_representation);
WHERE
  wr1: SIZEOF(SELF.items) >= 1;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT ((
    'CAST_PARTS_SCHEMA.FACE_SURFACE' IN TYPEOF(it)) OR (
    'CAST_PARTS_SCHEMA.ORIENTED_FACE' IN TYPEOF(it)))) )) = 0;
END_ENTITY; -- face_shape_representation
(*)
face_shape_representation_relationship
*)
ENTITY face_shape_representation_relationship
  SUBTYPE OF (representation_relationship);
WHERE
  wr1: 'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(SELF.
    rep_1);
  wr2: 'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(SELF.
    rep_2);
END_ENTITY; -- face_shape_representation_relationship
(*)
face_surface
*)
ENTITY face_surface
  SUBTYPE OF (face, geometric_representation_item);
  face_geometry : surface;

```

```

        same_sense      : BOOLEAN;
    WHERE
        wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_SURFACE' IN TYPEOF(
            face_geometry));
    END_ENTITY; -- face_surface
(*)
faceted_brep
*)
    ENTITY faceted_brep
        SUBTYPE OF (manifold_solid_brep);
    END_ENTITY; -- faceted_brep
(*)
feature_component_definition
*)
    ENTITY feature_component_definition
        SUBTYPE OF (characterized_object);
    WHERE
        wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
                SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE'))
                = 1)) )) = 0;
    END_ENTITY; -- feature_component_definition
(*)
feature_component_relationship
*)
    ENTITY feature_component_relationship
        SUPERTYPE OF (ONEOF (pattern_omit_membership,pattern_offset_membership))
        SUBTYPE OF (shape_aspect_relationship);
    WHERE
        wr1: ((SIZEOF(['CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT',
            'CAST_PARTS_SCHEMA.REPLICATE_FEATURE',
            'CAST_PARTS_SCHEMA.TRANSITION_FEATURE',
            'CAST_PARTS_SCHEMA.MODIFIED_PATTERN'] * TYPEOF(SELF.
            relating_shape_aspect)) = 1) OR (
            'CAST_PARTS_SCHEMA.FEATURE_DEFINITION' IN TYPEOF(SELF.
            relating_shape_aspect.of_shape.definition))) OR (
            'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
            SELF.relating_shape_aspect.of_shape.definition));
    END_ENTITY; -- feature_component_relationship
(*)
feature_definition
*)
    ENTITY feature_definition
        SUBTYPE OF (characterized_object);
    WHERE
        wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
            SELF) | (
                'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF(pdr.used_representation)) )) <= 1;
        wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
                SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) )) = 1)) )) = 0)) )) = 0;
wr3: SIZEOF(['CAST_PARTS_SCHEMA.BOSS',
'CAST_PARTS_SCHEMA.TURNED_KNURL', 'CAST_PARTS_SCHEMA.THREAD',
'CAST_PARTS_SCHEMA.GEAR', 'CAST_PARTS_SCHEMA.MARKING',
'CAST_PARTS_SCHEMA.RIB_TOP', 'CAST_PARTS_SCHEMA.ROUND_HOLE',
'CAST_PARTS_SCHEMA.OUTSIDE_PROFILE',
'CAST_PARTS_SCHEMA.POCKET',
'CAST_PARTS_SCHEMA.REMOVAL_VOLUME',
'CAST_PARTS_SCHEMA.REVOLVED_PROFILE',
'CAST_PARTS_SCHEMA.OUTER_ROUND',
'CAST_PARTS_SCHEMA.FLAT_FACE', 'CAST_PARTS_SCHEMA.PROTRUSION',
'CAST_PARTS_SCHEMA.ROUNDED_END', 'CAST_PARTS_SCHEMA.SLOT',
'CAST_PARTS_SCHEMA.SPHERICAL_CAP', 'CAST_PARTS_SCHEMA.STEP',
'CAST_PARTS_SCHEMA.COMPOUND_FEATURE',
'CAST_PARTS_SCHEMA.REPLICATE_FEATURE',
'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_FEATURE_DEFINITION'] *
TYPEOF(SELF)) = 1;
wr4: (NOT (SIZEOF(['CAST_PARTS_SCHEMA.ROUND_HOLE',
'CAST_PARTS_SCHEMA.BOSS', 'CAST_PARTS_SCHEMA.OUTSIDE_PROFILE',
'CAST_PARTS_SCHEMA.REMOVAL_VOLUME',
'CAST_PARTS_SCHEMA.FLAT_FACE', 'CAST_PARTS_SCHEMA.POCKET',
'CAST_PARTS_SCHEMA.PROTRUSION', 'CAST_PARTS_SCHEMA.RIB_TOP',
'CAST_PARTS_SCHEMA.ROUNDED_END', 'CAST_PARTS_SCHEMA.SLOT',
'CAST_PARTS_SCHEMA.STEP'] * TYPEOF(SELF)) = 1)) OR (SIZEOF(
QUERY ( pdr <* get_property_definition_representations(SELF)
| (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit')) )) >= 0);
END_ENTITY; -- feature_definition
(*)
feature_pattern
*)
ENTITY feature_pattern
SUBTYPE OF (replicate_feature);
WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
used_representation.items | (NOT (
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(srwp_i))) )) > 0)) ))
> 0)) = 0;
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
        'base feature placement')) ) > 1)) ) = 0)) ) = 0;
    END_ENTITY; -- feature_pattern
    (*
feeder
    *)
    ENTITY feeder
        SUBTYPE OF (gating_design_feature);
        WHERE

    (* feeders to connection_transition (as transition)    *)

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name IN['edge round'])) ) = 1 ))))=1;

    wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name IN [ 'fillet'])) ) = 1 ))))=1;

    (* ----- feeder to item_size (as shape) ----- *)

    WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='feeder occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
    AND
        (sar.description = 'feeder size usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
        IN TYPEOF(sar.relatng_shape_aspect))
        )) = 1 )))) >=1;

    (* ----- feeders to risers (as runner_connection) ----- *)

```

```

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='feeder occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser reference usage') AND
  ('CAST_PARTS_SCHEMA.RISER' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 )))) >=1;
END_ENTITY; -- feeder
(*)
fillet
*)
ENTITY fillet
  SUBTYPE OF (transition_feature);
  WHERE
    wr1: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
    wr2: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | ((NOT (SIZEOF(impl_rep.
      used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
      used_representation.items) <= 3)) )) = 0)) )) = 0);
    wr3: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
      = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) )) = 0);
    wr4: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELf,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'first offset')) ) <= 1)) ) = 0)) ) =
0);
wr5: (NOT (SELF.description = 'constant radius')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'second offset')) ) <= 1)) ) = 0)) )
= 0);
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') | (
NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'fillet face')) ) = 1)) ) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') | (
NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'first face shape')) ) = 1)) ) = 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') | (
NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
used_representation)) AND (pdr.used_representation.name =
'second face shape')) ) = 1)) ) = 0;
END_ENTITY; -- fillet
(*
filter
*)
ENTITY filter
SUBTYPE OF (gating_design_feature);
WHERE
(* ----- type_of_filter----- *)

```

```

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |

```

1520 © ISO 2008—All rights reserved



```

(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'filter type'))))=1 ))=1;

(* ----- filter to item_size (as filter_shape) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='filer occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))
AND
  (sar.description = 'filter size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE' IN
  TYPEOF(sar.relating_shape_aspect))
  )) =1 ))) >=1;

END_ENTITY; -- filter
(*
finishing_and_machining_operation
*)
ENTITY finishing_and_machining_operation
  SUBTYPE OF (customer_casting_requirement);
  WHERE
  (* ----- to descriptive_parameter(as comments) ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  (it.name = 'comments')
  ) )) =1 )))=0));

(* ----- to machining_activity (as finishing_operations) ----- *)

wr2:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  ((aam\action_method_assignment.role.name='finishing operation') AND
  ('CAST_PARTS_SCHEMA.MACHINING_ACTIVITY'
  IN TYPEOF(aam\action_method_assignment.assigned_action_method))))=1);

(* ----- to machining_activity (as machining_operations) ----- *)

wr3:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  ((aam\action_method_assignment.role.name='machining operation') AND
  ('CAST_PARTS_SCHEMA.MACHINING_ACTIVITY'
  IN TYPEOF(aam\action_method_assignment.assigned_action_method))))=1);

END_ENTITY; -- finishing_and_machining_operation
(*

```

**finishing\_equipment**

```
*)
ENTITY finishing_equipment
  SUBTYPE OF (casting_equipment);
END_ENTITY; -- finishing_equipment
```

(\*  
**first\_article**

```
*)
ENTITY first_article
  SUBTYPE OF (casting_verification);
WHERE
  (* ----- number of pieces -----*)
```

```
wr1: (NOT (SIZEOF (QUERY (ars <* USEDIN (SELF,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
  (NOT (SIZEOF (QUERY (ap <* USEDIN (ars.method,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (NOT (SIZEOF (QUERY (apr <* USEDIN (ap,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
  (SIZEOF (QUERY (it <* apr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF (it\measure_with_unit.value_component))) AND
  (it.name = 'number of pieces'))
  )) =1 ))) =0 )))=0 ))) =0));
```

```
(* --- first_article to approval (as customer_approval) ----- *)
```

```
wr2: SIZEOF (QUERY (aaa <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |
  (NOT (SIZEOF (QUERY (ra <* USEDIN (aaa,
  'CAST_PARTS_SCHEMA.ROLE_ASSOCIATION.ITEM_WITH_ROLE') |
  (ra.role.name='customer approval'))=1 )))=0;
```

```
(* ----- first_article to date (as delivery_date) ----- *)
```

```
wr3: (SIZEOF (QUERY (ada <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
  (ada.role.name='delivery date'))=1);
```

```
(* --first_article to master_sample (as customer_approved) ----- *)
```

```
wr4: SIZEOF (QUERY (ada <* USEDIN (SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_ASSIGNMENT.ASSIGNED_ACTION_REQUEST') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT'
  IN TYPEOF (ada)) AND
  (NOT (SIZEOF (QUERY (ari <* ada.items |
  'CAST_PARTS_SCHEMA.MASTER_SAMPLE' IN TYPEOF (ari))) =1))))=0;
```

```
END_ENTITY; -- first_article
```

(\*  
**fixture\_tab**

```

*)
  ENTITY fixture_tab
    SUBTYPE OF (casting_design_feature);
  END_ENTITY; -- fixture_tab
(*)
flask
*)
  ENTITY flask
    SUBTYPE OF (sand_cast_design_feature);
  WHERE

(* ----- flask to descriptive_parameter (as type_of_flask) ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'flask type')))) =1 ))))=1;

(* ----- flask to numeric_parameter (as cope_height) ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'cope height')) ) ) =1 ))))=1;

(* ----- flask to numeric_parameter (as cope_height) ----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'drag height')) ) ) =1 ))))=1;

(* ----- flask to numeric_parameter (as flask_length) ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'flask length')) ) ) =1 ))))=1;

```

ISO 10303-223:2008 (E)

```
(* ----- flask to numeric_parameter (as flask width) ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'flask width')) ) =1 )))=1;

(* ----- flask to pin_center (as pin_center_definition) ----- *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='flask reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pin center reference usage') AND
  ('CAST_PARTS_SCHEMA.PIN_CENTER' IN TYPEOF(sar.relater_shape_aspect))
  )) <=1 ))) >=1;

(* ----- flask to sand_mould (as generates) ----- *)

wr7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='flask reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sand mould reference usage') AND
  ('CAST_PARTS_SCHEMA.SAND_MOULD' IN TYPEOF(sar.relater_shape_aspect))
  )) =1 ))) >=1;

END_ENTITY; -- flask
(*
flaskless
*)
ENTITY flaskless
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <*
get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'description')) =1 )))=1;

(* ----- flaskless to mould_box (as mould_made_by) ----- *)

Wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='flaskless reference occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
```

```
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'mould box reference usage') AND
('CAST_PARTS_SCHEMA.MOULD_BOX' IN TYPEOF(sar.relying_shape_aspect))
)) >=0 )))) >=1;
```

```
END_ENTITY; -- flaskless
```

```
(*
```

## **flat\_face**

```
*)
```

```
ENTITY flat_face
```

```
  SUBTYPE OF (feature_definition);
```

```
  WHERE
```

```
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'removal direction')) ) = 1)) )
      = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
      | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'course of travel occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND (
      'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
      sar))) ) | (((('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
      TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
      relating_shape_aspect.description = 'linear')) AND (sdr.name
      = 'course of travel')) ) = 1)) ) = 1)) ) = 1)) ) = 0;
    wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
      | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'removal boundary occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
      sar))) ) | (('CAST_PARTS_SCHEMA.LINEAR_PROFILE' IN TYPEOF(
      sdr.relying_shape_aspect)) AND (sdr.name =
      'removal boundary')) ) = 1)) ) = 1)) ) = 1)) ) = 0;
    wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
```

```

    | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
    description = 'enclosed boundary occurrence') AND (SIZEOF(
    QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | ((sar.description =
    'profile usage') AND (
    'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
    sar))) ) | ((SIZEOF([
    'CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE',
    'CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
    'CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
    'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
    relating_shape_aspect)) = 1) AND (sdr.relatng_shape_aspect.
    description = 'boundary')) ) = 1)) ) <= 1)) ) = 0;
wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
    SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
    TYPEOF(pdr.used_representation)) AND (pdr.
    used_representation.name = 'maximum feature limit')) ) >= 0;
wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
    AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
    'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
    AND (csa.name = 'uncut area')) AND (SIZEOF(QUERY ( sar <*
    csa.component_relationships | ((
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) AND (SIZEOF(['CAST_PARTS_SCHEMA.BOSS',
    'CAST_PARTS_SCHEMA.PROTRUSION'] * TYPEOF(sar.
    related_shape_aspect)) = 1)) ) = 1)) ) <= 1)) ) = 1;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'removal depth')) ) <= 1)) ) = 0)) )
    = 0;

```

END\_ENTITY; -- flat\_face

(\*

## **flatness\_tolerance**

\*)

ENTITY flatness\_tolerance

SUBTYPE OF (geometric\_tolerance);

WHERE

```

    wr1: NOT (('CAST_PARTS_SCHEMA.' +
    'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));

```

```

    END_ENTITY; -- flatness_tolerance
  (*
founded_item
  *)
  ENTITY founded_item;
  END_ENTITY; -- founded_item
  (*
functionally_defined_transformation
  *)
  ENTITY functionally_defined_transformation;
    name          : label;
    description    : OPTIONAL text;
  END_ENTITY; -- functionally_defined_transformation
  (*
gating_design_feature
  *)
  ENTITY gating_design_feature
    SUBTYPE OF (casting_feature_definition);
    WHERE

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation) ) ) = 1;

  (* ----- to Orientation (as placement) ----- *)

    wr2:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
      (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
        (it.name = 'orientation')))) =1 ))))=1;

    END_ENTITY; -- gating_design_feature
  (*
gating_system
  *)
  ENTITY gating_system
    SUBTYPE OF (gating_design_feature);
    WHERE

  (* ----- gating_system to ingates (as gating_components) ----- *)

    wr1:  SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
      ((sa.description='gating system occurrence') AND
      (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
      (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
      (sar.description = 'gating system components usage') AND
      ( SIZEOF(['CAST_PARTS_SCHEMA.INGATE',
        'CAST_PARTS_SCHEMA.RISER',

```

```

        'CAST_PARTS_SCHEMA.RUNNER',
        'CAST_PARTS_SCHEMA.SPRUE']
        * TYPEOF(sar.relatng_shape_aspect)) =1))) >=1 ))))>=1;

    END_ENTITY; -- gating_system
    (*
gear
    *)
    ENTITY gear
    SUBTYPE OF (feature_definition);
    WHERE
        wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                SIZEOF(QUERY ( pdr <* USEDIN(pd,
                    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                    | (((
                        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                        TYPEOF(pdr.used_representation)) AND (10 <= SIZEOF(pdr.
                            used_representation.items))) AND (SIZEOF(pdr.
                                used_representation.items) >= 13))) = 1) )) = 1;
        wr2 : (NOT (SELF\characterized_object.description IN [
            'straight bevel gear','helical bevel gear','spur gear',
            'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
                (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                    | (
                        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                            QUERY ( it <* impl_rep.used_representation.items | ((
                                SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                                    'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it))
                                    = 2) AND (it.name = 'number of teeth')) )) = 1)) )) = 0)) ))
                = 0);
        wr3 : (NOT (SELF\characterized_object.description IN [
            'straight bevel gear','helical bevel gear','spur gear',
            'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
                (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                    | (
                        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                            QUERY ( it <* impl_rep.used_representation.items | ((
                                SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                                    'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
                                    TYPEOF(it)) = 2) AND (it.name = 'reference pressure angle')) ))
                                    = 1)) )) = 0)) )) = 0);
        wr4 : (NOT (SELF\characterized_object.description IN [
            'straight bevel gear','helical bevel gear','spur gear',
            'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
                (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```



```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| (
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
  QUERY ( it <* impl_rep.used_representation.items | ((
  SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
  = 2) AND (it.name = 'rake shift factor')) ) = 1)) ) = 0)) )
= 0);
wr5 : (NOT (SELF\characterized_object.description IN [
  'straight bevel gear','helical bevel gear','spur gear',
  'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
  (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
  | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'nominal tooth depth')) ) = 1)) ) = 0)) )
    = 0);
wr6 : (NOT (SELF\characterized_object.description IN [
  'straight bevel gear','helical bevel gear','spur gear',
  'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
  (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
  | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'face width')) ) = 1)) ) = 0)) ) = 0);
wr7 : (NOT (SELF\characterized_object.description IN [
  'straight bevel gear','helical bevel gear','spur gear',
  'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
  (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
  | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'tip diameter')) ) = 1)) ) = 0)) ) =
    0);
wr8 : (NOT (SELF\characterized_object.description IN [
  'straight bevel gear','helical bevel gear','spur gear',
  'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
        | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'module or diametral pitch'))
        AND ((it.description = 'module') OR (it.description =
        'diametral pitch')))) = 1)) = 0)) = 0);
wr9 : (NOT (SELF\characterized_object.description IN [
        'straight bevel gear','helical bevel gear','spur gear',
        'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
        | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'internal or external gear'))
        AND ((it.description = 'internal') OR (it.description =
        'external')))) = 1)) = 0)) = 0);
wr10: (NOT (SELF\characterized_object.description IN [
        'straight bevel gear','helical bevel gear','spur gear',
        'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
        | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((
        SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'normal attribute')) = 1)) = 0)) = 0);
wr11: (NOT (SELF\characterized_object.description IN [
        'straight bevel gear','helical bevel gear','spur gear',
        'helical gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
        | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((
        SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'root fillet radius')) <= 1)) = 0)) = 0);

```

```

wr12: (NOT (SELF\characterized_object.description IN ['helix gear',
'helical bevel gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(
SELF, 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| (
'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | ((
SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'reference helix angle')) ))
= 1)) ) = 0)) ) = 0);

wr13: (NOT (SELF\characterized_object.description IN ['helix gear',
'helical bevel gear'])) OR (SIZEOF(QUERY ( pd <* USEDIN(
SELF, 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| (
'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'gear tooth')) AND ((it.
description = 'left hand tooth') OR (it.description =
'right hand tooth')))) ) = 1)) ) = 0)) ) = 0);

wr14: (NOT (SELF\characterized_object.description IN [
'straight bevel gear', 'helical bevel gear'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| (
'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | ((
SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'tip angle')) ) = 1)) ) =
0)) ) = 0);

wr15: (NOT (SELF\characterized_object.description IN [
'straight bevel gear', 'helical bevel gear'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
| (
'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
QUERY ( it <* impl_rep.used_representation.items | ((
SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'root angle')) ) = 1)) )

```

```

= 0)) )) = 0);
wr16: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
| ((sar.description = 'applied shape') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
relating_shape_aspect))) = 1) )) = 1)) )) = 0;
END_ENTITY; -- gear

```

(\*

### **general\_property**

\*)

```

ENTITY general_property;
    id          : identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- general_property

```

(\*

### **general\_property\_association**

\*)

```

ENTITY general_property_association;
    name          : label;
    description   : OPTIONAL text;
    base_definition : general_property;
    derived_definition : derived_property_select;
WHERE
    wr1: SIZEOF(USEDIN(derived_definition, ('CAST_PARTS_SCHEMA.' +
'GENERAL_PROPERTY_ASSOCIATION.') + 'DERIVED_DEFINITION')) =
        1;
    wr2: derived_definition.name = base_definition.name;
END_ENTITY; -- general_property_association

```

(\*

### **generic\_manufacturing\_resource**

\*)

```

ENTITY generic_manufacturing_resource
    SUBTYPE OF (action_resource);
WHERE
(* -----quantity -----*)
wr1:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELF) |
(('CAST_PARTS_SCHEMA.REPRESENTATION' IN
TYPEOF(pdr.representation)) AND
(SIZEOF(QUERY (it <* pdr.representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component)) AND
(it.name = 'resource quantity')) )) =1 ))))>=1;

```

```
(* -----unit -----*)

wr2:  SIZEOF(QUERY ( pdr <* get_resource_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'resource quantity')AND
  ('CAST_PARTS_SCHEMA.CONTEXT_DEPENDENT_UNIT'
  IN TYPEOF(it\measure_with_unit.unit_component)) AND
  (it\measure_with_unit.unit_component\conversion_based_unit.name='each')
  ) ) =1 ) ) ) >=1;
END_ENTITY; -- generic_manufacturing_resource
```

```
(*
```

### **geometric\_alignment**

```
*)
```

```
ENTITY geometric_alignment
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY; -- geometric_alignment
```

```
(*
```

### **geometric\_intersection**

```
*)
```

```
ENTITY geometric_intersection
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY; -- geometric_intersection
```

```
(*
```

### **geometric\_representation\_context**

```
*)
```

```
ENTITY geometric_representation_context
  SUBTYPE OF (representation_context);
  coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context
```

```
(*
```

### **geometric\_representation\_item**

```
*)
```

```
ENTITY geometric_representation_item
  SUPERTYPE OF (ONEOF (point,direction,vector,placement,
    cartesian_transformation_operator,curve,surface,edge_curve,
    face_surface,poly_loop,vertex_point,solid_model))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    wr1: SIZEOF(QUERY ( using_rep <* using_representations(SELf) | (NOT
      ('CAST_PARTS_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT' IN
      TYPEOF(using_rep.context_of_items))) ) ) = 0;
```

```

    END_ENTITY; -- geometric_representation_item
(*)
geometric_tolerance
*)
    ENTITY geometric_tolerance;
        name                : label;
        description          : text;
        magnitude            : measure_with_unit;
        toleranced_shape_aspect : shape_aspect;
    WHERE
        wr1: magnitude.value_component >= 0;
    END_ENTITY; -- geometric_tolerance
(*)
geometric_tolerance_relationship
*)
    ENTITY geometric_tolerance_relationship;
        name                : label;
        description          : text;
        relating_geometric_tolerance : geometric_tolerance;
        related_geometric_tolerance : geometric_tolerance;
    END_ENTITY; -- geometric_tolerance_relationship
(*)
geometric_tolerance_with_datum_reference
*)
    ENTITY geometric_tolerance_with_datum_reference
        SUBTYPE OF (geometric_tolerance);
        datum_system : SET [1:?] OF datum_reference;
    END_ENTITY; -- geometric_tolerance_with_datum_reference
(*)
geometric_tolerance_with_defined_unit
*)
    ENTITY geometric_tolerance_with_defined_unit
        SUBTYPE OF (geometric_tolerance);
        unit_size : measure_with_unit;
    WHERE
        wr1: unit_size.value_component > 0;
    END_ENTITY; -- geometric_tolerance_with_defined_unit
(*)
global_uncertainty_assigned_context
*)
    ENTITY global_uncertainty_assigned_context
        SUBTYPE OF (representation_context);
        uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
    END_ENTITY; -- global_uncertainty_assigned_context
(*)
global_unit_assigned_context
*)
    ENTITY global_unit_assigned_context
        SUBTYPE OF (representation_context);
        units : SET [1:?] OF unit;
    END_ENTITY; -- global_unit_assigned_context
(*)

```

**group**

```

*)
ENTITY group;
    name          : label;
    description   : OPTIONAL text;
DERIVE
    id : identifier := get_id_value(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- group

```

(\*

**group\_assignment**

```

*)
ENTITY group_assignment
    ABSTRACT SUPERTYPE;
    assigned_group : group;
DERIVE
    role : object_role := get_role(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- group_assignment

```

(\*

**group\_relationship**

```

*)
ENTITY group_relationship;
    name          : label;
    description   : OPTIONAL text;
    relating_group : group;
    related_group  : group;
END_ENTITY; -- group_relationship

```

(\*

**heat\_treat**

```

*)
ENTITY heat_treat
    SUBTYPE OF (non_casting_activity);
WHERE
    wr1: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
        IN TYPEOF(am)) AND
        (SIZEOF(QUERY( cp <* am.items |
        ('CAST_PARTS_SCHEMA.HEAT_TREAT_REQUIREMENT'
        IN TYPEOF(cp)))) =1 ))))=0));
END_ENTITY; -- heat_treat

```

(\*

**heat\_treat\_requirement**

```

*)
ENTITY heat_treat_requirement

```

ISO 10303-223:2008 (E)

```

SUBTYPE OF (process_requirement);
WHERE
(* ----- additional_requirements ----- *)

wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'additional requirement')
) )) =1 )))=0));

(* ----- softening_anneal_required ----- *)

wr2:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'softening anneal required') AND
(it.description IN (['TRUE','FALSE']))
) )) =1 )))=0));

(* ----- stress_relief_required ----- *)

wr3:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'stress relief required') AND
(it.description IN (['TRUE','FALSE']))
) )) =1 )))=0));

(* --- to organization (as where_heat_treat_is_to_be_performed----- *)

wr4:(SIZEOF(QUERY ( aam <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
((aam\organization_assignment.role.name='activity organization id') AND
('CAST_PARTS_SCHEMA.ORGANIZATION'
IN TYPEOF(aam\organization_assignment.assigned_organization))))=1);

(* ----- to process_property (as heat_treatment_per_design_data) ----- *)

wr5:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION'
IN TYPEOF(pdr.related_property_definition)) AND
(pdr.related_property_definition.name='process property'))
)=1);

(* -----to specification (as heat_treat_specification) ----- *)

wr6:SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |

```



```

(adr.assigned_document.kind.product_data_type=
'heat_treat_specification') ))
=1;

(* ---- to mechanical_property_requirements (as mechanical_property) ---- *)

wr7:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.MECHANICAL_PROPERTY_REQUIREMENTS'
IN TYPEOF(pdr.related_property_definition))
))=1);

END_ENTITY; -- heat_treat_requirement
(*
hole_bottom
*)
ENTITY hole_bottom
SUBTYPE OF (shape_aspect);
WHERE
wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
SELF.of_shape.definition);
wr2 : SELF.description IN ['through','flat','flat with radius',
'flat with taper','spherical','conical'];
wr3 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 0)) ) )
= 0);
wr4 : (NOT (SELF.description IN ['flat with radius',
'flat with taper','spherical','conical'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) = 1)) ) ) = 0);
wr5 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 0)) ) ) = 0)) ) ) = 0);
wr6 : (NOT (SELF.description IN ['flat with radius','spherical']))
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1)) ) = 0)) ) = 0);
wr7 : (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 2)) ) = 0)) ) = 0);
wr8 : (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | ((NOT (SIZEOF(impl_rep.
used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
used_representation.items) <= 2)) ) = 0)) ) = 0);
wr9 : (SELF.description = 'through') OR (SIZEOF(QUERY ( fcr <*
QUERY ( sar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | ((sar.description =
'hole bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (fcr.name IN ['hole depth start',
'hole depth end']) ) ) >= 1);
wr10: (NOT (SELF.description = 'flat with radius')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'corner radius')) ) = 1)) ) = 0)) )
= 0);
wr11: (NOT (SELF.description = 'spherical')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'radius')) ) = 1)) ) = 0)) ) = 0);
wr12: (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'tip radius')) ) <= 1)) ) = 0)) ) =
0);
wr13: (NOT (SELF.description = 'conical')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'tip angle')) ) = 1)) ) =
0)) ) = 0);
wr14: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | ((sar.description =
'hole bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
'bottom condition occurrence') AND (
'CAST_PARTS_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
related_shape_aspect.of_shape.definition))) AND (
'CAST_PARTS_SCHEMA.HOLE_BOTTOM' IN TYPEOF(fcr.
relating_shape_aspect))) ) ) >= 1;
wr15: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'final diameter')) ) = 1)) ) = 0)) )
= 0);
wr16: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'taper diameter')) ) = 1)) )
= 0)) ) = 0);
END_ENTITY; -- hole_bottom
(*)
hyperbola
*)
ENTITY hyperbola
  SUBTYPE OF (conic);
  semi_axis      : positive_length_measure;
  semi_imag_axis : positive_length_measure;
END_ENTITY; -- hyperbola
(*)
id_attribute
*)
ENTITY id_attribute;
  attribute_value : identifier;
  identified_item : id_attribute_select;
END_ENTITY; -- id_attribute
(*)
identification_assignment
*)
ENTITY identification_assignment
  ABSTRACT SUPERTYPE;
  assigned_id : identifier;
  role       : identification_role;
END_ENTITY; -- identification_assignment
(*)
identification_role
*)
ENTITY identification_role;
  name      : label;

```

```

        description : OPTIONAL text;
    END_ENTITY; -- identification_role
    (*
illustration
    *)
    ENTITY illustration
        SUBTYPE OF (document);
    WHERE
        wr1: SIZEOF(QUERY ( aduc <* USEDIN(SELF,
            'CAST_PARTS_SCHEMA.'+
            'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS') |
            ('CAST_PARTS_SCHEMA.VIEW_REFERENCE'
            IN TYPEOF
            (aduc\document_usage_constraint_assignment.assigned_document_usage)
            )) >= 0;

```

```

    END_ENTITY; -- illustration
    (*

```

### **impregnation\_activity**

```

    *)
    ENTITY impregnation_activity
        SUBTYPE OF (non_casting_activity);
    END_ENTITY; -- impregnation_activity
    (*

```

### **in\_facility\_location**

```

    *)
    ENTITY in_facility_location
        SUBTYPE OF (representation);
    WHERE
        wr1: (NOT(SIZEOF(QUERY ( it <* SELF.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'building or area')) )) =0));

        wr2: (NOT(SIZEOF(QUERY ( it <* SELF.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'location code')) )) =0));

        wr3: (NOT(SIZEOF(QUERY ( it <* SELF.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'sublocation')) )) =0));

    END_ENTITY; -- in_facility_location
    (*

```

### **ingate**

```

    *)
    ENTITY ingate
        SUBTYPE OF (gating_design_feature);
    WHERE
    (* ----- cope_or_drag ----- *)

    WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN

```

```

        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
          (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
              ((it.description IN ['cope', 'drag','cope and drag']
                )) ) ) ) =1 ) ) )>=1;

(* ----- ingates to ingate_contacts (as contacts) ----- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
          (sar.description = 'ingate contact reference usage') AND
            ('CAST_PARTS_SCHEMA.INGATE_CONTACT'
              IN TYPEOF(sar.relateing_shape_aspect))
          )) >=1 )))) >=1;

(* ----- ingates to item_size (as ingate_shape) ----- *)

WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
      AND
        (sar.description = 'ingate size usage') AND
          ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
            IN TYPEOF(sar.relateing_shape_aspect))
          )) <=1 )))) >=1;

END_ENTITY; -- ingate
(*
ingate_contact
*)
ENTITY ingate_contact
  SUBTYPE OF (gating_design_feature);
  WHERE
(* ----- cope_or_drag ----- *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
      (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
          ((it.description IN ['cope', 'drag','cope and drag']
            )) ) ) ) =1 ) ) )>=1;

(* - ingate_contacts to connection_transition (as runner_transition) -- *)

wr2 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |

```

```

(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'runner transition edge round')) ) ) <=1 ))))=1;

wr3 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'runner transition fillet')) ) ) <=1 ))))=1;

(* - ingate_contacts to connection_transition (as pattern_transition) - *)

wr4 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'pattern transition edge round')) ) ) <=1 ))))=1;

wr5 :SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'pattern transition edge fillet')) ) ) <=1 ))))=1;

(* -ingate_contacts to production_pattern_definition (as applied_to) - *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='ingate contact occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production pattern definition reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

(* ---- ingates_contacts to runner (as runner_connection) ----- *)

wr7:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |

```

```

        ((sa.description='ingate contact occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'runner reference usage') AND
        ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(sar.relatng_shape_aspect))
        )) >=1 )))) >=1;

(* -- ingate_contacts to shape_element (as break_off_connection)----- *)

WR8:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='ingate contact occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'ingate contact shape usage') AND
        ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
        )) >=1 )))) >=1;

END_ENTITY; -- ingate_contact
(*
inspection_activity
*)
ENTITY inspection_activity
    SUBTYPE OF (casting_activity);
    WHERE
    (* ----- inspectrion_level ----- *)

wr1:    SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'inspection level')) )) =1 ))))=1;

(* ----- reason_for_inspection ----- *)

wr2:    SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
        TYPEOF(pdr.representation)) AND
        (SIZEOF(QUERY (it <* pdr.representation.items |
        (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'reason for inspection') AND
        (it.description IN [
        'initial inspectrion',
        'engineering change',
        'tooling replacement',
        'tooling transfer',
        'tooling refurbishment',
        'discrepancy correction',

```



```

'material or supplier source change',
'production sample set']])) =1 ))))>=1;

(* ----- type_of_inspection ----- *)

wr3:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
      (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'type of inspection') AND
        (it.description IN [
          'destructive',
          'dimensional',
          'functional testing',
          'gauging',
          'hardness property',
          'internal property',
          'leak testing',
          'material composition',
          'surface','visual']])) =1 ))))>=1;

(* -----nspection_activity to approval (as foundry approval) ----- *)

wr4: (NOT(SIZEOF(QUERY(aaa <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |
  (SIZEOF(QUERY(ra <* USEDIN(aaa,
  'CAST_PARTS_SCHEMA.ROLE_ASSOCIATION.ITEM_WITH_ROLE') |
  (ra.role.name='foundry approval')) =1 )) )=0));

(* -- to casting_verification (as quality_assurance) ----- *)

wr5: SIZEOF(QUERY(aaa <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
  ('CAST_PARTS_SCHEMA.CASTING_VERIFICATION'
  IN TYPEOF(aaa.request))))=1;

(* --- to part_version (as inspected_part) ----- *)

wr6: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(am)) AND
  (SIZEOF(QUERY( cp <* am.items |
  ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(cp)))) =1 ))))=0));

(* -- to product_quality_report (as inspection_report) ----- *)

wr7: (('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (NOT (SIZEOF(QUERY( doc <* SELF.documents |
  (doc.kind.product_data_type='product quality report'))=0)));

```

ISO 10303-223:2008 (E)

```
(* ----- to inspection_plan (as customer_requirements) ----- *)

wr8: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    (SIZEOF(QUERY ( ip <* USEDIN(ap.related_method,
        'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD') |
        'CAST_PARTS_SCHEMA.INSPECTION_PLAN' IN TYPEOF(ip)))=1)))=0));

    END_ENTITY; -- inspection_activity
    (*
inspection_or_test_requirement
    *)
    ENTITY inspection_or_test_requirement
        SUBTYPE OF (customer_casting_requirement);
        WHERE
    (* ----- validation_id ----- *)

    wr1:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
            (it.name = 'validation id')
            ) ) =1 ) )=0));

    (* ----- test name ----- *)

    wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
            (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
            (it.name = 'test name')
            ) )=1 ) ) ) = 0);

    (* ----- to numeric_parameter(as frequency)----- *)

    wr3:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
            ( ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
            ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
            IN TYPEOF(it\measure_with_unit.value_component))) AND
            (it.name = 'frequency'))=1)))=0);

    (* ----- to numeric_parameter(as sampling_run_size)----- *)

    wr4:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
            ( ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
            ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
            IN TYPEOF(it\measure_with_unit.value_component))) AND
            (it.name = 'sampling run size'))=1)))=0);
```

```
(* ----- to customer_casting_requirement (as inspection_against)----- *)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT'
  IN TYPEOF(pdr.related_property_definition)) )) = 1;

(* ---- to specification (as production_quality_requirement_standard)--- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.description =
  'production quality requirement standard') ))
  = 1;

  END_ENTITY; -- inspection_or_test_requirement
(*
inspection_or_test_result
*)
ENTITY inspection_or_test_result
  SUBTYPE OF (property_definition);
  WHERE
(* ----- result ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'result')
  ) )=1 )))) = 0);

(* -- inspection_or_test_result to checking_aid_tool (as aid_used) --- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.CHECKING_AID_TOOL'
  IN TYPEOF(pdr.related_property_definition))
  ))=1);

(* --inspection_or_test_result to shape_aspect(as applies_to)----- *)

wr3: (NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ( ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE'
  IN TYPEOF(pdr.related_property_definition)) AND
  (SIZEOF(QUERY( sa <* USEDIN(pdr.related_property_definition,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sa))
  )) =1 )) )) =0));
```

```
(* -- inspection_or_test_result to sampled_set(as belongs_to)----- *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.SAMPLED_SET'
  IN TYPEOF(pdr.related_property_definition)
  )) =1);
```

```
END_ENTITY; -- inspection_or_test_result
(*)
```

## inspection\_plan

```
(*)
ENTITY inspection_plan
  SUBTYPE OF (product_definition_process);
  WHERE
  (* ----- to descriptive_parameter(as corrective_action) ----- *)
```

```
wr1: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'test name') AND
  (it.description IN ['weld', 'impregnation'])))) =1 ))))>=1;
```

```
(* -----to descriptive_parameter(as correction_procedures) ----- *)
```

```
wr2: SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'correction_procedures') AND
  (it.description IN ['dimensional', 'visual', 'internal']
  ))) =1 ))))>=1;
```

```
(* ----- to inspection_or_test_requirement(as customer_requirements) ---- *)
```

```
wr3: (SIZEOF(QUERY(ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
  ('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_REQUIREMENT'
  IN TYPEOF(ap.property_or_shape)
  )))=1);
```

```
(* ----- to special_inspection_requirement(as customer_requirements) ---- *)
```

```
wr4: (SIZEOF(QUERY(ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
  ('CAST_PARTS_SCHEMA.SPECIAL_INSPECTION_REQUIREMENT'
  IN TYPEOF(ap.property_or_shape)
  )))=1);
```

```

END_ENTITY; -- inspection_plan
(*)
instanced_feature
*)
ENTITY instanced_feature
  SUBTYPE OF (feature_definition, shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(SELF.of_shape.
      definition);
    wr2: SELF.product_definitional;
END_ENTITY; -- instanced_feature
(*)
integration_interval
*)
ENTITY integration_interval
  SUBTYPE OF (action_method);
  WHERE
    wr1: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
      (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
      (NOT (SIZEOF(QUERY(eds <* ama.items |
      (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
      (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY(mri <* pd.used_representation.items |
      ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(mri)) = 2)AND
      (mri.name = 'delta time'))
      )) =1 ))) =0 )))))=0 )))))=0))););

    wr2: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
      (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
      (NOT (SIZEOF(QUERY(eds <* ama.items |
      (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
      (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY(mri <* pd.used_representation.items |
      ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(mri)) = 2)AND
      (mri.name = 'length of elapsed time interval'))
      )) =1 ))) =0 )))))=0 )))))=0))););

  END_ENTITY; -- integration_interval
(*)
investment_casting_master
*)

```

ISO 10303-223:2008 (E)

```

ENTITY investment_casting_master
  SUBTYPE OF (investment_casting_tooling);
  WHERE

(* ----- shrink factor for master tooling ----- *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor for master tooling')) ) = 1 )))=1;

END_ENTITY; -- investment_casting_master
(*
investment_casting_tooling
*)
ENTITY investment_casting_tooling
  SUBTYPE OF (casting_product_definition);
  WHERE

(* to investment_casting_tooling_definition (as investment_tooling) *** *)
(* --- to cast_part_with_rigging (as prototype) -- *)

wr1: ((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='investment casting tooling occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description =
  'investment casting tooling definition reference usage')AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_CASTING_TOOLING_DEFINITION'
  IN TYPEOF(sar.relater_shape_aspect))
  )) >=1 ))) >=1) OR
  (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING'
  IN TYPEOF(pdr.related_product_definition)) AND
  (pdr.name='prototype'))
  ))>=1));

(* ***** material_definitions ***** *)

wr2: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='tooling material') AND
  (SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |

```

```

    (mfuo.name IN [
    'plaster',
    'expandable polystyrene',
    'iron',
    'soft wood',
    'shell',
    'aluminum',
    'epoxy',
    'wood metal reinforced',
    'brass',
    'hard wood',
    'urethane',
    'other',
    'steel',
    'ceramic',
    'polymethylmethacrylate',
    'plywood'] )
    )) >=1 )) )) =0 ));

(* ***** tooling_identifer ***** *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'tooling_identifer')))) =1 ))))=1;

END_ENTITY; -- investment_casting_tooling
(*
investment_casting_tooling_definition
*)
ENTITY investment_casting_tooling_definition
  SUBTYPE OF (investment_design_feature);
WHERE

(* -- to investment_mould (as tree_structure) ----- *)

WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='investment casting tooling definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment mould reference usage') AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_MOULD'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) <=1 )))) >=1;

(* ----- to sprue_and_runner_mould (as metal_flow_system) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='investment casting tooling definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,

```

ISO 10303-223:2008 (E)

```

        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'sprue and runner mould reference usage') AND
        ('CAST_PARTS_SCHEMA.SPRUE_AND_RUNNER_MOULD'
        IN TYPEOF(sar.relying_shape_aspect))
        )) =1 )))) >=1;

(* ----- to production_core_box (as core tooling) ----- *)

WR3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='investment casting tooling definition occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'production core box reference usage') AND
        ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
        IN TYPEOF(sar.relying_shape_aspect))
        )) >=0 )))) >=1;

(* ----- to production_investment_cast_mould (as die_mould) -----
*)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='investment casting tooling definition occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'production investment cast mould reference usage')
AND
        ( SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD',
        'CAST_PARTS_SCHEMA.LOST_FOAM_CASTING_DIE']
        * TYPEOF(sar.relying_shape_aspect)) =1))) >=0 ))))>=1;

(* ----- to assembly (as pattern_section_assembly) ----- *)

wr5: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
        (('CAST_PARTS_SCHEMA.ASSEMBLY'
        IN TYPEOF(adr.assigned_action_method)) )) =1);

(* ----- to shape_element (as pattern_section_definition)----- *)

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='investment casting tooling definition occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'pattern section shape usage') AND
        ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
        )) >=1 )))) >=1;

        END_ENTITY; -- investment_casting_tooling_definition
(*

```



**investment\_design\_feature**

```

*)
ENTITY investment_design_feature
  SUBTYPE OF (casting_feature_definition);
WHERE

WR1: SIZEOF(QUERY( pdr <* get_property_definition_representations (SELF) |
  'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
  IN TYPEOF ( pdr.used_representation ) ) ) =1;

(* ***** investment_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 ))))=1;

END_ENTITY; -- investment_design_feature

```

**investment\_mould**

```

*)
ENTITY investment_mould
  SUBTYPE OF (investment_design_feature);
WHERE

(* ----- number_of_impressions ----- *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of patterns')) ) ) =1 ))))=1;

(* ----- mould material ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'mould material')))) =1 ))))=1;

END_ENTITY; -- investment_mould

```

**known\_source**

```

*)
ENTITY known_source

```

```

        SUBTYPE OF (external_source, pre_defined_item);
    END_ENTITY; -- known_source
(*)
length_measure_with_unit
*)
    ENTITY length_measure_with_unit
        SUBTYPE OF (measure_with_unit);
        WHERE
            wr1: 'CAST_PARTS_SCHEMA.LENGTH_UNIT' IN TYPEOF(SELF\
                measure_with_unit.unit_component);
    END_ENTITY; -- length_measure_with_unit
(*)
length_unit
*)
    ENTITY length_unit
        SUBTYPE OF (named_unit);
        WHERE
            wr1: ((((((SELF\named_unit.dimensions.length_exponent = 1) AND (SELF\
                \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
                named_unit.dimensions.time_exponent = 0)) AND (SELF\
                named_unit.dimensions.electric_current_exponent = 0)) AND (
                SELF\named_unit.dimensions.
                thermodynamic_temperature_exponent = 0)) AND (SELF\
                named_unit.dimensions.amount_of_substance_exponent = 0)) AND
                (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
    END_ENTITY; -- length_unit
(*)
library_class_version_assignment
*)
    ENTITY library_class_version_assignment
        SUBTYPE OF (applied_external_identification_assignment);
    END_ENTITY; -- library_class_version_assignment
(*)
library_property_version_assignment
*)
    ENTITY library_property_version_assignment
        SUBTYPE OF (applied_external_identification_assignment);
    END_ENTITY; -- library_property_version_assignment
(*)
limits_and_fits
*)
    ENTITY limits_and_fits;
        form_variance : label;
        zone_variance : label;
        grade          : label;
        source         : text;
    END_ENTITY; -- limits_and_fits
(*)
line
*)
    ENTITY line

```

```

SUBTYPE OF (curve);
  pnt : cartesian_point;
  dir : vector;
WHERE
  wr1: dir.dim = pnt.dim;
END_ENTITY; -- line
(*)
line_profile_tolerance
*)
ENTITY line_profile_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: (NOT (('CAST_PARTS_SCHEMA.' +
      'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
      OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
        datum_system) <= 3);
    wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF\geometric_tolerance.
      toleranced_shape_aspect, 'CAST_PARTS_SCHEMA.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
      name IN ['affected plane association',
      'resulting intersection curve association'])) ) = 1;
  END_ENTITY; -- line_profile_tolerance

```

```

(*)
linear_profile
*)
ENTITY linear_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
      .of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation))) ) = 1)) ) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) ) = 0)) ) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) )) = 1)) )) = 0)) )) = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'profile length')) )) = 1)) )) = 0)) ))
= 0;
END_ENTITY; -- linear_profile
(*)
local_time
*)
ENTITY local_time;
    hour_component      : hour_in_day;
    minute_component    : OPTIONAL minute_in_hour;
    second_component    : OPTIONAL second_in_minute;
    zone                : coordinated_universal_time_offset;
WHERE
    wr1: valid_time(SELF);
END_ENTITY; -- local_time
(*)
location_shape_representation
*)
ENTITY location_shape_representation
    SUBTYPE OF (shape_representation);
WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
        'CAST_PARTS_SCHEMA.POINT' IN TYPEOF(it))) )) = 0;
END_ENTITY; -- location_shape_representation
(*)
loop
*)
ENTITY loop
    SUPERTYPE OF (ONEOF (vertex_loop,edge_loop,poly_loop))
    SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop
(*)
loose_piece
*)
ENTITY loose_piece
    SUBTYPE OF (sand_cast_design_feature);
WHERE

```

```

wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN ['aluminum','cast iron','urethane','wood']))) ) = 1;

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='loose piece shape occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

```

```
END_ENTITY; -- loose_piece
```

```
(*
```

### lost\_foam\_casting\_die

```
*)
```

```
ENTITY lost_foam_casting_die
  SUBTYPE OF (production_tool);
WHERE
```

```
(* ***** raw bead size ***** *)
```

```

wr1:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) ) |
  (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'raw bead size')
  )))=1))))=0))))=0);

```

```
(* ***** steam_line_location ***** *)
```

```

wr2:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
  IN TYPEOF(pdr.used_representation)) ) |
  (NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'steam line location')
  )))>=1))))=0))))=0);

```

```
(* ***** pattern_removal_instruction ***** *)
```

```

wr3:(SIZEOF(QUERY ( pd <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
(('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'pattern removal instruction'
)))>=1))))=0))))=0);

(* -----die_wall_thickness----- *)

wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
(it.name = 'die wall thickness')) ) ) >= 1)) ) = 0)) ) = 0;

(* ----- to planar_element (as steam_line_location)----- *)

wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) AND
(pdr.used_representation.name = 'steam line location'))
)) >= 1)) ) = 1;

(* ----- to planar_element (as fill_gun_location)----- *)

wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) AND
(pdr.used_representation.name = 'fill gun location')) ) ) >= 1)) ) = 1;

(* ----- item_size (as fill_gun_size)----- *)

wr7: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'fill gun size usage') AND

```

```

(('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))) |
(('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
 IN TYPEOF(sdr.relating_shape_aspect) )
)) ) >=1 )) )) )=1 )) ))=0);

(* ----- to cooling port (as mould cooling) ----- *)

WR8: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'cooling port reference usage') AND
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
(('CAST_PARTS_SCHEMA.COOLING_PORT'
 IN TYPEOF(sdr.relating_shape_aspect))
)) ) =1 )) )) )=1 )) ))=0);

(* ----- to die_clamping (as stationary_die) ----- *)

WR9: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
((sar.description = 'die clamping reference usage') AND
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
(('CAST_PARTS_SCHEMA.DIE_CLAMPING'
 IN TYPEOF(sdr.relating_shape_aspect))
)) ) =1 )) )) )=1 )) ))=0);

(* ----- to machine_mounting (as stationary die mounting) ----- *)

WR10: SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
( 'CAST_PARTS_SCHEMA.MACHINE_MOUNTING'
 IN TYPEOF ( pdr.used_representation ) ))) =1;

(* ----- to flask (as defines flask type) ----- *)

WR11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
((sa_occ.description = 'lost foam casting die occurrence') AND
( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |

```

ISO 10303-223:2008 (E)

```

((sar.description = 'flask reference usage') AND
 ('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) |
 ('CAST_PARTS_SCHEMA.FLASK' IN TYPEOF(sdr.relater_shape_aspect))
)) ) =1 )) ) ) =1 )) ) =0);

(* ----- to gating_design_feature (as gating requirements) ----- *)

WR12: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
 ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
 (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
 ((sa_occ.description = 'lost foam casting die occurrence') AND
 ( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 ((sar.description = 'gating design feature reference usage') AND
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)))) |
 (('CAST_PARTS_SCHEMA.GATING_DESIGN_FEATURE'
 IN TYPEOF(sdr.relater_shape_aspect))
)) ) >=1 )) ) ) =1 )) ) =0);

(* ----- to slide or die_clamping (as movable die item) ----- *)

WR13: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
 ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) ) |
 (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
 ((sa_occ.description =
 'lost foam casting die occurrence') AND
 ( (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 ((sar.description =
 'slide or die clamping reference usage') AND
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP')
 IN TYPEOF(sar))) ) |
 ( SIZEOF(['CAST_PARTS_SCHEMA.SLIDE',
 'CAST_PARTS_SCHEMA.DIE_CLAMPING']
 * TYPEOF(sdr.relater_shape_aspect)) = 1)
)) ) >=1 )) ) ) >=1 )) ) =0);

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr14: (SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
 ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
 IN TYPEOF ( pdr.used_representation ) ))) =1) OR
 (SIZEOF(QUERY(ada <* USEDIN(SELF,
 'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
 (ada.role.name='tool stored'))=1);
END_ENTITY; -- lost_foam_casting_die

```

**lot\_effectivity**



```

*)
  ENTITY lot_effectivity
    SUBTYPE OF (effectivity);
    effectivity_lot_id : identifier;
    effectivity_lot_size : measure_with_unit;
  END_ENTITY; -- lot_effectivity
(*)
machine
*)
  ENTITY machine
    SUBTYPE OF (action_resource);
  END_ENTITY;
(*)
machining_process
*)
  ENTITY machining_process
    SUBTYPE OF (manufacturing_process);
  END_ENTITY;
(*)
machine_setup
*)
  ENTITY machine_setup
    SUBTYPE OF (process_plan_activity);
  END_ENTITY;
(*)
machine_allowance
*)
  ENTITY machine_allowance
    SUBTYPE OF (property_definition);
    WHERE
    wr1: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY ( it <* pdr.used_representation.items |
      ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
      (it.name = 'allowance value')) = 1) )) = 1) ) = 1;

    wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.'+
      'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
      ('CAST_PARTS_SCHEMA.INSTANCED_FEATURE'
      IN TYPEOF(pdr.related_property_definition.definition)) )) = 1;

  END_ENTITY; -- machine_allowance
(*)
machine_element_relationship
*)
  ENTITY machine_element_relationship
    SUBTYPE OF (action_resource_relationship);
    WHERE
    wr1: 'CAST_PARTS_SCHEMA.MACHINE' IN TYPEOF(SELF.related_resource);

```

```

        wr2: SIZEOF(['CAST_PARTS_SCHEMA.EJECTOR_EQUIPMENT',
                    'CAST_PARTS_SCHEMA.CONTROLLER'] * TYPEOF(SELF.
                    relating_resource)) = 1;
    END_ENTITY; -- machine_element_relationship
(*)
machine_mounting
*)
    ENTITY machine_mounting
        SUBTYPE OF (representation);
        WHERE

(*) ----- machine_mounting_type ----- *)

    wr1: SIZEOF(QUERY ( it <* SELF.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'machine mounting type')
        ))) =1;

(*) ----- material_type ----- *)

    wr2: SIZEOF(QUERY ( it <* SELF.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'material type')
        ))) =1;

(*) ----- machine_mounting to equipment_size (as dimensions) ----- *)

    wr3: SIZEOF(QUERY ( it <* SELF.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'diameter'))
        ))=1;

    wr4: SIZEOF(QUERY ( it <* SELF.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'draft'))
        ))=1;

    wr5: SIZEOF(QUERY ( it <* SELF.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'length'))
        ))=1;

    wr6: SIZEOF(QUERY ( it <* SELF.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND (it.name = 'width'))
        ))=1;

```

```

wr7: SIZEOF(QUERY ( it <* SELF.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND (it.name = 'height'))
  ))=1;

(* ---- machine_mounting to location_element (as location) ---- *)

wr8: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
  IN TYPEOF(ap.rep_2)) ))=1;

(* ---- mount_with ---- *)

wr9: SIZEOF(QUERY ( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'mount with') AND
  (it.description IN ['clamps', 'holes', 'slots'])
  )) >=1;

END_ENTITY; -- machine_mounting
(*
machining_activity
*)
ENTITY machining_activity
  SUBTYPE OF (non_casting_activity);
  WHERE
(* ---- to external_schema_definition (as iso_10303_240_data) ----- *)

wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (SIZEOF(QUERY ( edi <* sar.items |
  ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA'
  IN TYPEOF(edi)) )) <= 1))))>=0;

END_ENTITY; -- machining_activity
(*
make_from_usage_option
*)
ENTITY make_from_usage_option
  SUBTYPE OF (product_definition_usage);
  ranking : INTEGER;
  ranking_rationale : text;
  quantity : measure_with_unit;
  WHERE
  wr1: (NOT ('NUMBER' IN TYPEOF(quantity.value_component))) OR (
  quantity.value_component > 0);
END_ENTITY; -- make_from_usage_option
(*)

```

### **manifold\_solid\_brep**

```
*)
  ENTITY manifold_solid_brep
    SUBTYPE OF (solid_model);
    outer : closed_shell;
  END_ENTITY; -- manifold_solid_brep
(*
```

### **manufacturing\_activity**

```
*)
  ENTITY manufacturing_activity
    SUBTYPE OF (process_plan_activity);
  END_ENTITY;
(*
```

### **manufacturing\_activity\_relationship**

```
*)
  ENTITY manufacturing_activity_relationship
    SUBTYPE OF (sequential_method);
  WHERE

    wr1: ('CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS'
          IN TYPEOF(SELF.relying_method));

    wr2: ('CAST_PARTS_SCHEMA.MANUFACTURING_ACTIVITY'
          IN TYPEOF(SELF.related_method));

  END_ENTITY;
(*
```

### **manufacturing\_process**

```
*)
  ENTITY manufacturing_process
    SUBTYPE OF (action_method);
  WHERE
    (* - manufacturing_process to casting_activity (as assigned_operation) *)

  wr1:  SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP'
    IN TYPEOF(ap)) AND
    ('CAST_PARTS_SCHEMA.CASTING_ACTIVITY'
    IN TYPEOF(ap.related_method))))=1;

    (* ----- manufacturing_process to setup_activity (as setup) ----- *)

  wr2: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    (('CAST_PARTS_SCHEMA.SINGLE_ACTIVITY_RELATIONSHIP' IN TYPEOF(ap)) AND
    (SIZEOF(['CAST_PARTS_SCHEMA.ANCILLARY_SETUP',
    'CAST_PARTS_SCHEMA.MACHINE_SETUP'] * TYPEOF(ap.related_method)) = 1))
    )=0));

  END_ENTITY; -- manufacturing_process
(*
```

**manufacturing\_process\_relationship**

```

*)
ENTITY manufacturing_process_relationship
  SUBTYPE OF (sequential_method);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
      related_method);
    wr2: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
      relating_method);
  END_ENTITY; -- manufacturing_process_relationship

```

(\*

**mapped\_item**

```

*)
ENTITY mapped_item
  SUBTYPE OF (representation_item);
  mapping_source : representation_map;
  mapping_target : representation_item;
  WHERE
    wr1: acyclic_mapped_representation(using_representations(SELF), [SELF]);
  END_ENTITY; -- mapped_item

```

(\*

**marking**

```

*)
ENTITY marking
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) AND ((2 <= SIZEOF(pdr.
      used_representation.items)) AND (SIZEOF(pdr.
      used_representation.items) <= 6))) ) = 1) ) = 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (it.name = 'marking text')) ) = 1)) ) = 0)) )
      = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +

```

```

        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'special instructions')) ) <= 1)) )
        = 0)) ) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'font name')) ) <= 1)) ) = 0)) )
        = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'character height')) ) <= 1)) ) = 0)) )
        = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'character spacing')) ) <= 1)) ) = 0)) )
        = 0;
wr7: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +

```

```

        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'applied_shape') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
        relating_shape_aspect))) = 1) )) = 1)) )) = 0;
    END_ENTITY; -- marking
    (*
mass_measure_with_unit
    *)
    ENTITY mass_measure_with_unit
        SUBTYPE OF (measure_with_unit);
        WHERE
            wr1: 'CAST_PARTS_SCHEMA.MASS_UNIT' IN TYPEOF(SELF\measure_with_unit.
                unit_component);
    END_ENTITY; -- mass_measure_with_unit
    (*
mass_unit
    *)
    ENTITY mass_unit
        SUBTYPE OF (named_unit);
        WHERE
            wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
                \named_unit.dimensions.mass_exponent = 1)) AND (SELF\
                named_unit.dimensions.time_exponent = 0)) AND (SELF\
                named_unit.dimensions.electric_current_exponent = 0)) AND (
                SELF\named_unit.dimensions.
                thermodynamic_temperature_exponent = 0)) AND (SELF\
                named_unit.dimensions.amount_of_substance_exponent = 0)) AND
                (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
    END_ENTITY; -- mass_unit
    (*
master_pattern
    *)
    ENTITY master_pattern
        SUBTYPE OF (sand_casting_tooling);
        WHERE

        wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
            ('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF(pdr.used_representation)) )) = 1;

    (* ----- shrink factor for master pattern ----- *)

    WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
            TYPEOF(pdr.used_representation)) AND
            (SIZEOF(QUERY (it <* pdr.used_representation.items |
                (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
                    IN TYPEOF(it)) AND (it.name = 'shrink factor for master pattern')
                ))) = 1 )))) = 1;

```

ISO 10303-223:2008 (E)

```
(* ** to machining_allowance (as master_allowance_on_master_pattern) * *)

wr3: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION' ) |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))=1);

END_ENTITY; -- master_pattern
(*
master_sample
*)
ENTITY master_sample
  SUBTYPE OF (cast_part);
  WHERE
(* ----- master_sample to date (as production_date) *)

wr1: SIZEOF(QUERY(da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS' ) |
  ((da.role.name='production date') AND
  ('CAST_PARTS_SCHEMA.CALENDAR_DATE'
  IN TYPEOF(da\date_assignment.assigned_date))))=1;

(* --- master_sample to design_part (as product_specification_used) -- *)

wr2: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATED_PRODUCT_DEFINITION' ) |
  ('CAST_PARTS_SCHEMA.DESIGN_PART'
  IN TYPEOF(pdr.relating_product_definition)) )=1;

(* ----- master_sample to heat (as belongs_to_heat) -- *)

wr3: SIZEOF(QUERY(da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS' ) |
  ('CAST_PARTS_SCHEMA.DATED_EFFECTIVITY'
  IN TYPEOF(da\effectivity_assignment.assigned_effectivity)
  ))=1;

(* ----- master_sample to lot (as belongs_to_lot) -- *)

wr4: SIZEOF(QUERY(da <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS' ) |
  ('CAST_PARTS_SCHEMA.LOT_EFFECTIVITY'
  IN TYPEOF(da\effectivity_assignment.assigned_effectivity)
  ))=1;

(* ----- master_sample to process_plan (as process_plan_used) ----*)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.DEFINED_PRODUCT' ) |
  ((pdr.description='process plan used') AND
  ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(pdr.process)) )
  )<=1;
```



```

WR6: SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.EXTERNAL_SOURCE'
  IN TYPEOF(pdr\external_identification_assignment.source)) AND
  (NOT (SIZEOF(QUERY(es <* USEDIN(pdr.source,
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_ITEM.SOURCE') |
  'CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_SCHEMA' IN TYPEOF(es)
  )) <=1 )) )) )=0;

```

```
END_ENTITY; -- master_sample
```

```
(*
```

### **material\_designation**

```
*)
```

```

ENTITY material_designation;
  name          : label;
  definitions   : SET [1:?] OF characterized_definition;
END_ENTITY; -- material_designation

```

```
(*
```

### **material\_property**

```
*)
```

```

ENTITY material_property
  SUBTYPE OF (property_definition);
  UNIQUE
  url : name, definition;
  WHERE
  wr1: ('CAST_PARTS_SCHEMA.CHARACTERIZED_OBJECT' IN TYPEOF(SELF\
  property_definition.definition)) OR (SIZEOF(bag_to_set(
  USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
  QUERY ( temp <* bag_to_set(USEDIN(SELF,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
  'CAST_PARTS_SCHEMA.' +
  'MATERIAL_PROPERTY_REPRESENTATION') IN TYPEOF(temp)) )) = 0);
END_ENTITY; -- material_property

```

```
(*
```

### **material\_property\_representation**

```
*)
```

```

ENTITY material_property_representation
  SUBTYPE OF (property_definition_representation);
  dependent_environment : data_environment;
END_ENTITY; -- material_property_representation

```

```
(*
```

### **material\_structure**

```
*)
```

```

ENTITY material_structure
  SUBTYPE OF (material_property);
  WHERE

```

```
(* --- material_structure to material_structure_element (as element)-- *)
```

## ISO 10303-223:2008 (E)

```
WR1: (SELF.definition.frame_of_reference.name=
      'material structure definition');

(* --- material_structure to illustration (as illustration)----- *)

wr2: SIZEOF(QUERY (adr <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
      ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
      (adr.assigned_document))))=1;

(* --- material_structure to specification (as referenced_to)----- *)

wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
      ('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
      (adr.assigned_document))))=1;

      END_ENTITY; -- material_structure
(*
material_usage
*)
      ENTITY material_usage
      SUBTYPE OF (property_definition);
      WHERE
(* ----- Material_usage to numeric_parameter (as amount_used) ----- *)

wr1: (NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY ( it <* pdr.used_representation.items |
      ( (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
      (it.name = 'amount used')) ) ) =1 ) ) =0));

(* ----- material_usage to material (as material_used) -----*)

wr2: ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (SELF.definition));

      END_ENTITY; -- material_usage
(*
material_usage_record
*)
      ENTITY material_usage_record
      SUBTYPE OF (activity_result_record);
      WHERE
(* ----- Material_usage to numeric_parameter (as amount_used) ----- *)

wr1: (SIZEOF(QUERY ( it <* SELF.items |
      ( (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
      (it.name = 'actual substance usage amount')) ) ) =1);

(* ----- material_usage_record to material_usage (as recorded_usage)----- *)
```

```

wr2: (SIZEOF(QUERY( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_REPRESENTATION.USED_REPRESENTATION') |
    ('CAST_PARTS_SCHEMA.MATERIAL_USAGE' IN TYPEOF(pdr.definition))))=1);

END_ENTITY; -- material_usage_record
(*)
measure_qualification
*)
ENTITY measure_qualification;
    name          : label;
    description    : text;
    qualified_measure : measure_with_unit;
    qualifiers     : SET [1:?] OF value_qualifier;
WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
        'CAST_PARTS_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp) ) )
        < 2;
END_ENTITY; -- measure_qualification
(*)
measure_representation_item
*)
ENTITY measure_representation_item
    SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item
(*)
measure_with_unit
*)
ENTITY measure_with_unit
    SUPERTYPE OF (ONEOF (length_measure_with_unit, mass_measure_with_unit,
        time_measure_with_unit, plane_angle_measure_with_unit,
        ratio_measure_with_unit));
    value_component : measure_value;
    unit_component  : unit;
WHERE
    wr1: valid_units(SELF);
END_ENTITY; -- measure_with_unit
(*)
mechanical_property_requirements
*)
ENTITY mechanical_property_requirements
    SUBTYPE OF (property_definition);
WHERE
(*) ----- chemical_analysis_certification ----- *)

wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'chemical analysis certification') AND

```

ISO 10303-223:2008 (E)

```

        (it.description IN (['TRUE','FALSE']))
        ) ) =1 ))=0));

(* ----- mechanical_properties_certification ----- *)

wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'mechanical_properties_certification') AND
(it.description IN (['TRUE','FALSE']))
) ) =1 ))=0));

(* ----- tensile_strength ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'tensile_strength')
))) =1 ))=0));

(* ----- yeild_strength ----- *)

wr4: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'yeild_strength')
))) =1 ))=0));

(* ----- to material_hardness (as hardness)----- *)

wr5: SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.MATERIAL_PROPERTY'
IN TYPEOF(pdr.related_property_definition)) ) ) = 1;

(* -----to specification (as requirement_specification) ----- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
(adr.assigned_document.kind.product_data_type=
'requirement_specification') ) )
= 1;

END_ENTITY; -- mechanical_property_requirements

```

```

(*)
mechanical_stress
*)
  ENTITY mechanical_stress
    SUBTYPE OF (customer_casting_requirement);
    WHERE

  (* ----- to descriptive_parameter(as type_of_stress) ----- *)

  wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND
    (it.name = 'type of stress')
    ) ) = 1 )))=0));

  (* ----- to numeric_parameter(as maximum_psi_design_stress) ----- *)

  wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'maximum psi design stress'))
    ) ) = 1 )))=0));

  (* ----- to numeric_parameter(as percent_safety_factor_required) ----- *)

  wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'percent safety factor required'))
    ) ) = 1 )))=0));

  END_ENTITY; -- mechanical_stress
(*)
melting_equipment
*)
  ENTITY melting_equipment
    SUBTYPE OF (casting_equipment);
  END_ENTITY; -- melting_equipment
(*)
meshing_condition
*)
  ENTITY meshing_condition
    SUBTYPE OF (action_method);
    WHERE

```

```

(* number_of_elements *)

wr1:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
(NOT (SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
(NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(mri <* pd.used_representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(mri\measure_with_unit.value_component))AND
(mri.name = 'number of elements'))
)) =1 ))) =0 )))))=0 )))))=0));

(* number_of_nodes *)

wr2:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
(NOT (SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
(NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(mri <* pd.used_representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(mri\measure_with_unit.value_component))AND
(mri.name = 'number of nodes'))
)) =1 ))) =0 )))))=0 )))))=0));

(* meshing_condition to numeric_range(as density_of_nodes) *)

wr3:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
(NOT (SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
(NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY(mri <* pd.used_representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
TYPEOF(mri)) = 2))AND
(mri.name = 'density of nodes'))
)) =1 ))) =0 )))))=0 )))))=0));

(* meshing_condition to numeric_range(as density_of_elements) *)

wr4:(NOT(SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |

```

```

(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
 (NOT (SIZEOF(QUERY(eds <* ama.items |
 ('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
 (NOT (SIZEOF(QUERY(pd <* USEDIN(eds,
 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 (SIZEOF(QUERY(mri <* pd.used_representation.items |
 ((SIZEOF([
 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
 'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
 TYPEOF(mri)) = 2)AND
 (mri.name = 'density of elements'))
 )) =1 ))) =0 )))))=0 )))))=0));

(* meshing_condition to shape_aspect (as area_to_be_meshed) *)

wr5:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
 'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
 (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama)) AND
 (SIZEOF(QUERY(eds <* ama.items |
 ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
 )) =1 )) ))=0));

END_ENTITY; -- meshing_condition
*)
metalcaster_simulation
*)
ENTITY metalcaster_simulation
  SUBTYPE OF (simulation_process);
  WHERE
(* ----- to simulation_exception_report (as defect_results) -----*)

wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc <* SELF.documents |
  (doc.kind.product_data_type='simulation exception report'))=1);

(* ----- to tooling_report (as suggestion_for_changes) -----*)

wr2: ('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELF)) AND
  (SIZEOF(QUERY( doc <* SELF.documents |
  (doc.kind.product_data_type='simulation tooling report'))=1);

(* ---- to gating_system (as proposed_rigging) ----- *)

wr3:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM' IN TYPEOF(eds.definition))

```

```

    ))) =1 )))=0));

(* ----- to pattern_equipment_available (as available_tooling) ----- *)

wr4: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PATTERN_EQUIPMENT_AVAILABLE' IN TYPEOF (eds))
    ))) <=1 )))=0));

END_ENTITY; -- metalcaster_simulation

```

(\*

### **mismatch\_tolerance**

```

*)
ENTITY mismatch_tolerance
  SUBTYPE OF (property_definition);
  WHERE
(* ----- to numeric_parameter(as maximum_amount) ----- *)

```

```

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component))) AND
    (it.name = 'maximum amount')
    )) =1 )) ) = 0);

```

```

(* ----- to parting_surface(as with_respect_to)----- *)

```

```

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.'+
    'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
    (NOT(SIZEOF(QUERY(ps <* USEDIN(pdr.related_property_definition,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    'CAST_PARTS_SCHEMA.PARTING_SURFACE' IN TYPEOF(ps)))=1)))=0);

```

```

END_ENTITY; -- mismatch_tolerance

```

(\*

### **modified\_geometric\_tolerance**

```

*)
ENTITY modified_geometric_tolerance
  SUBTYPE OF (geometric_tolerance);
  modifier : limit_condition;
END_ENTITY; -- modified_geometric_tolerance

```

(\*

### **modified\_pattern**

```

*)
ENTITY modified_pattern
  SUBTYPE OF (shape_aspect);

```



```

WHERE
  wr1: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) ) | ((SIZEOF([
    'CAST_PARTS_SCHEMA.REPLICATE_FEATURE',
    'CAST_PARTS_SCHEMA.INSTANCED_FEATURE'] * TYPEOF(fcr.
    related_shape_aspect.of_shape.definition)) >= 1) AND (fcr.
    description = 'base shape')) ) = 1;
  wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) ) | ((SIZEOF([
    'CAST_PARTS_SCHEMA.CIRCULAR_PATTERN',
    'CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(fcr.
    related_shape_aspect.of_shape.definition)) = 1) AND (fcr.
    description = 'base pattern')) ) = 1;
  wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (SIZEOF(QUERY ( msar <* USEDIN(
    sar.related_shape_aspect,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | ((SIZEOF([
    'CAST_PARTS_SCHEMA.PATTERN_OFFSET_MEMBERSHIP',
    'CAST_PARTS_SCHEMA.PATTERN_OMIT_MEMBERSHIP'] * TYPEOF(sar))
    = 1) AND (sar.description = 'modified pattern')) AND (sar
    :<>: msar)) ) >= 1) ) = 0;
END_ENTITY; -- modified_pattern
(*
mould_box
*)
ENTITY mould_box
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
  (* ----- mould_box to item_size (as box_size) ----- *)

  WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='mould box occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
  AND
    (sar.description = 'mould box dimension usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect)
    )) =1 ))) >=1;

  (* ***** mould_box to descriptive_parameter (as description) ***** *)

  wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN

```

```

        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
          (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'mould box description')))) <=1 )))=1;

    END_ENTITY; -- mould_box
  (*
mould_lock
  *)
    ENTITY mould_lock
      SUBTYPE OF (sand_cast_design_feature);
      WHERE
  (* ----- mould_locks to item_size (as lock_dimensions) ----- *)

    WR1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
      ((sa.description='mould lock occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
          'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
            (sar.description = 'mould lock dimension usage') AND
              ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
                IN TYPEOF(sar.relatng_shape_aspect))
                )) =1 )))) >=1;

  (* ***** mould_locks to descriptive_parameter (as description) ***** *)

    wr2:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
          (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'mould lock description')))) <=1
      )))=1;

    END_ENTITY; -- mould_lock
  (*
moulding_equipment
  *)
    ENTITY moulding_equipment
      SUBTYPE OF (casting_equipment);
    END_ENTITY; -- moulding_equipment
  (*
name_attribute
  *)
    ENTITY name_attribute;
      attribute_value : label;
      named_item      : name_attribute_select;
    END_ENTITY; -- name_attribute
  (*
named_unit
  *)
    ENTITY named_unit

```

```

    SUPERTYPE OF (ONEOF (si_unit,conversion_based_unit,
        context_dependent_unit) ANDOR ONEOF (length_unit,mass_unit,
        time_unit,plane_angle_unit,solid_angle_unit,ratio_unit));
    dimensions : dimensional_exponents;
END_ENTITY; -- named_unit
(*)
next_assembly_usage_occurrence
*)
ENTITY next_assembly_usage_occurrence
    SUBTYPE OF (assembly_component_usage);
END_ENTITY; -- next_assembly_usage_occurrence
(*)
ngon_closed_profile
*)
ENTITY ngon_closed_profile
    SUBTYPE OF (shape_aspect);
WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
        .of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) | (NOT ((SIZEOF(impl_rep.
        used_representation.items) >= 3) AND (SIZEOF(impl_rep.
        used_representation.items) <= 4))) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
        used_representation.items | (((((srwp_i.name = 'orientation')
        OR (srwp_i.name = 'number of sides')) OR (srwp_i.name =
        'circumscribed diameter') OR (srwp_i.name = 'corner radius'))
        OR (srwp_i.name = 'diameter across flats')) )) = SIZEOF(pdr.
        used_representation.items))) )) = 1)) )) = 1;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
        'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component))) AND (it.name =
        'number_of_sides')) )) = 1)) )) = 0)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name IN ['circumscribed diameter',
        'diameter across flats'])) )) = 1)) )) = 0)) )) = 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'corner radius')) )) <= 1)) )) = 0)) ))
= 0;
END_ENTITY; -- ngon_closed_profile
(*
non_casting_activity
*)
ENTITY non_casting_activity
    SUBTYPE OF (casting_activity);
    WHERE
(* ----- to descriptive_parameter (as operation description) ----- *)

```

```

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'operation description')
    )))) =1 ))))>=1;

(* ---- to finishing_or_machining_operations (as requirement) ----- *)

wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(sar)) AND
    (SIZEOF(QUERY ( edi <* sar.items |
    ('CAST_PARTS_SCHEMA.FINISHING_AND_MACHINING_OPERATION'
    IN TYPEOF(edi)) )) <= 1))))>=0;

(* --- to cast_part (as part_geometry_with_machining_allowance) ----- *)
(* --- to design_part (as part_geometry_with_no_machining_allowance) -- *)

wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(sar)) AND
    (NOT(SIZEOF(QUERY ( ip <* sar.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.CAST_PART',
    'CAST_PARTS_SCHEMA.DESIGN_PART'] * TYPEOF(ip)) <=1))
    ) =0 ))))>=0);

    END_ENTITY; -- non_casting_activity
(*
non_casting_equipment_process
*)
    ENTITY non_casting_equipment_process
        SUBTYPE OF (non_machining_process);
    END_ENTITY; -- non_casting_equipment_process
(*
non_machining_process
*)
    ENTITY non_machining_process
        SUBTYPE OF (manufacturing_process);
    END_ENTITY; -- non_machining_process
(*
non_permanent_moulding_process
*)
    ENTITY non_permanent_moulding_process
        SUBTYPE OF (casting_activity);
        WHERE

```

ISO 10303-223:2008 (E)

```
(* ----- moulding_process ----- *)

wr1: (SELF.description IN ['shell','green sand','no bake',
                           'lost foam','vacuum']);

(* ----- number_of_impressions ----- *)

wr2:    SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
                    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
                     TYPEOF(pdr.representation)) AND
                     (SIZEOF(QUERY (it <* pdr.representation.items |
                    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
                     ('COUNT_PARTS_SCHEMA.COUNT_MEASURE'
                     IN TYPEOF(it\measure_with_unit.value_component)) AND
                     (it.name = 'number of impressions')) ) =1 ))))=1;

(* ----- pattern_form ----- *)

wr3:    SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
                    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
                     TYPEOF(pdr.representation)) AND
                     (SIZEOF(QUERY (it <* pdr.representation.items |
                    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
                     IN TYPEOF(it)) AND (it.name = 'pattern form') AND
                     (it.description IN [
                     'mounted match plate',
                     'mounted cope and drag inserts',
                     'mounted cope and drag boards',
                     'mounted flaskless cope and drag',
                     'loose split pattern',
                     'loose pattern with follow block',
                     'sweep'])))) =1 ))))>=1;

(* ---- non_permanent_moulding_process to flask (as flask_type) ----- *)
(* ---- non_permanent_moulding_process to flaskless (as flask_type) ---- *)
(* ---- non_permanent_moulding_process to mould_box (as flask_type) ---- *)

wr4: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
    (SIZEOF(['CAST_PARTS_SCHEMA.FLASK',
    'CAST_PARTS_SCHEMA.FLASKLESS'] * TYPEOF(eds.definition)) =1
    ))) =1 )))) =0));

(* ---- to production_pattern_definition (as pattern) ----- *)

wr5: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
```

```

IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
(SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION',
'CAST_PARTS_SCHEMA.LOST_FOAM_CASTING_DIE',
'CAST_PARTS_SCHEMA.INVESTMENT_MOULD'] * TYPEOF(eds.definition)
=1)
))) =1 )))) =0));

```

```
END_ENTITY; -- non_permanent_moulding_process
```

```
(*
```

## object\_role

```
*)
```

```

ENTITY object_role;
    name          : label;
    description    : OPTIONAL text;
END_ENTITY; -- object_role

```

```
(*
```

## open\_path\_profile

```
*)
```

```

ENTITY open_path_profile
SUBTYPE OF (shape_aspect);
WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
        .of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
        'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
        pdr.used_representation)) AND (pdr.used_representation.name
        = 'profile limit')) )) <= 1)) )) = 0;
END_ENTITY; -- open_path_profile
(*)
open_shell
*)
ENTITY open_shell
  SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell
(*)
ordered_part
*)
ENTITY ordered_part
  SUBTYPE OF (action_assignment, characterized_object);
  items : SET [1:?] OF ordered_item;
  WHERE
wr1: (SIZEOF(USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION')) = 1);

wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
  = 0);

wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF( pdr.used_representation.items) = 1)) )) = 0)) )) = 0);

wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
  'CAST_PARTS_SCHEMA.' +
  'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'quantity')) )) = 1)) ))
  = 0)) )) = 0);

END_ENTITY; -- ordered_part

```



(\*)

**organization**

\*)

```

ENTITY organization;
    id          : OPTIONAL identifier;
    name        : label;
    description : OPTIONAL text;
END_ENTITY; -- organization

```

(\*)

**organization\_assignment**

\*)

```

ENTITY organization_assignment
    ABSTRACT SUPERTYPE;
    assigned_organization : organization;
    role                  : organization_role;
END_ENTITY; -- organization_assignment

```

(\*)

**organization\_role**

\*)

```

ENTITY organization_role;
    name : label;
DERIVE
    description : text := get_description_value(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- organization_role

```

(\*)

**organizational\_address**

\*)

```

ENTITY organizational_address
    SUBTYPE OF (address);
    organizations : SET [1:?] OF organization;
    description   : OPTIONAL text;
END_ENTITY; -- organizational_address

```

(\*)

**organizational\_project**

\*)

```

ENTITY organizational_project;
    name                : label;
    description          : OPTIONAL text;
    responsible_organizations : SET [1:?] OF organization;
DERIVE
    id : identifier := get_id_value(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- organizational_project

```

(\*)

**oriented\_closed\_shell**

\*)

© ISO 2008 — All rights reserved

1585

```

ENTITY oriented_closed_shell
  SUBTYPE OF (closed_shell);
  closed_shell_element : closed_shell;
  orientation           : BOOLEAN;
  DERIVE
    SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                          conditional_reverse(SELF.
                                          orientation,SELF.
                                          closed_shell_element.cfs_faces);

  WHERE
    wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_CLOSED_SHELL' IN TYPEOF(SELF.
      closed_shell_element));
END_ENTITY; -- oriented_closed_shell

```

(\*)

**oriented\_edge**

\*)

```

ENTITY oriented_edge
  SUBTYPE OF (edge);
  edge_element : edge;
  orientation  : BOOLEAN;
  DERIVE
    SELF\edge.edge_start : vertex := boolean_choose(SELF.orientation,
      SELF.edge_element.edge_start,SELF.
      edge_element.edge_end);
    SELF\edge.edge_end   : vertex := boolean_choose(SELF.orientation,
      SELF.edge_element.edge_end,SELF.
      edge_element.edge_start);

  WHERE
    wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_EDGE' IN TYPEOF(SELF.
      edge_element));
END_ENTITY; -- oriented_edge

```

(\*)

**oriented\_face**

\*)

```

ENTITY oriented_face
  SUBTYPE OF (face);
  face_element : face;
  orientation  : BOOLEAN;
  DERIVE
    SELF\face.bounds : SET [1:?] OF face_bound := conditional_reverse(
      SELF.orientation,SELF.face_element.bounds);

  WHERE
    wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_FACE' IN TYPEOF(SELF.
      face_element));
END_ENTITY; -- oriented_face

```

(\*)

**oriented\_open\_shell**

\*)

```

ENTITY oriented_open_shell
  SUBTYPE OF (open_shell);
  open_shell_element : open_shell;
  orientation        : BOOLEAN;

```

```

DERIVE
  SELF\connected_face_set.cfs_faces : SET [1:?] OF face :=
                                          conditional_reverse(SELF.
                                          orientation,SELF.
                                          open_shell_element.cfs_faces);

WHERE
  wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_OPEN_SHELL' IN TYPEOF(SELF.
    open_shell_element));
END_ENTITY; -- oriented_open_shell
(*)
oriented_path
*)
ENTITY oriented_path
  SUBTYPE OF (path);
  path_element : path;
  orientation : BOOLEAN;
DERIVE
  SELF\path.edge_list : LIST [1:?] OF UNIQUE oriented_edge :=
                                          conditional_reverse(SELF.orientation,SELF.
                                          path_element.edge_list);

WHERE
  wr1: NOT ('CAST_PARTS_SCHEMA.ORIENTED_PATH' IN TYPEOF(SELF.
    path_element));
END_ENTITY; -- oriented_path
(*)
oriented_surface
*)
ENTITY oriented_surface
  SUBTYPE OF (surface);
  orientation : BOOLEAN;
END_ENTITY; -- oriented_surface
(*)
other_inspection
*)
ENTITY other_inspection
  SUBTYPE OF (inspection_or_test_result);
WHERE
  (* ----- inspection result ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION' ) |
  (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'inspection result')
  ) )=1 )))) = 0);

(* -- to inspection_or_test_requirement (as requirements)----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION' ) |

```

```

('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_REQUIREMENT'
IN TYPEOF(pdr.related_property_definition)
)) =1);

```

```

END_ENTITY; -- other_inspection

```

```

(*)

```

## **outer\_round**

```

*)

```

```

ENTITY outer_round

```

```

  SUBTYPE OF (feature_definition);

```

```

  WHERE

```

```

    wr1: (NOT (SELF\characterized_object.description = 'outer diameter'))
      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.
        used_representation.items) = 3)) ) = 1) ) = 1);

```

```

    wr2: (NOT (SELF\characterized_object.description =
      'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND ((2 <= SIZEOF(pdr.
        used_representation.items)) AND (SIZEOF(pdr.
        used_representation.items) <= 3))) ) = 1) ) = 1);

```

```

    wr3: SELF\characterized_object.description IN ['outer diameter',
      'outer diameter to shoulder'];

```

```

    wr4: (NOT (SELF\characterized_object.description = 'outer diameter'))

```

```

      OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
      = 2) AND (it.name = 'length')) ) = 1)) ) = 0)) ) = 0);

```

```

    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([

```

```

'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'diameter')) ) = 1)) ) = 0)) ) = 0;
wr6: (NOT (SELF\characterized_object.description =
'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'v-shape boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.VEE_PROFILE' IN TYPEOF(sdr.
relating_shape_aspect)) AND (sdr.relating_shape_aspect.
description = 'v-shape')) ) = 1)) ) = 1)) ) = 0);
wr7: (NOT (SELF\characterized_object.description = 'outer diameter'))
OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'reduced size occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
AND ('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(sdr
relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.OUTER_ROUND' IN TYPEOF(sdr.
related_shape_aspect.of_shape.definition))) AND (sdr.name =
'reduced size')) ) = 1)) ) <= 1)) ) = 0);
wr8: (NOT (SELF\characterized_object.description =
'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'length')) ) <= 1)) ) = 0)) ) = 0);
wr9: (NOT (SELF\characterized_object.description =
'outer diameter to shoulder')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (

```

```

        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'feature length')) ) <= 1)) ) = 0)) )
        = 0);
    END_ENTITY; -- outer_round
(*)
outside_profile
*)
    ENTITY outside_profile
        SUBTYPE OF (feature_definition);
        WHERE
            wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                TYPEOF(pdr.used_representation)) AND (SIZEOF(pdr.
                used_representation.items) = 1)) ) = 1) ) = 1;
            wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
                | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
                description IN ['boundary occurrence',
                'non-planar boundary occurrence',
                'partial circular boundary occurrence',
                'closed circular boundary occurrence',
                'open rectangular boundary occurrence',
                'closed rectangular boundary occurrence']) ) = 1)) ) = 0;
            wr3 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
                | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
                description = 'boundary occurrence') ) = 1)) ) = 0)) OR (
                SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
                | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
                QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
                'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND (('CAST_PARTS_SCHEMA.' +
                'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
                SIZEOF(['CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE',

```

```

'CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE',
'CAST_PARTS_SCHEMA.SQUARE_U_PROFILE',
'CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
'CAST_PARTS_SCHEMA.ROUNDED_U_PROFILE',
'CAST_PARTS_SCHEMA.LINEAR_PROFILE',
'CAST_PARTS_SCHEMA.VEE_PROFILE',
'CAST_PARTS_SCHEMA.TEE_PROFILE',
'CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
relating_shape_aspect)) = 1) AND (sdr.relating_shape_aspect
.description = 'outside boundary')) ) = 1) )) = 0);
wr4 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description IN ['complex boundary occurrence',
'partial circular boundary occurrence',
'closed circular boundary occurrence',
'open rectangular boundary occurrence',
'closed rectangular boundary occurrence']) ) = 1)) )) = 0))
OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (NOT (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile floor usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.PROFILE_FLOOR' IN
TYPEOF(sdr.relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.OUTSIDE_PROFILE' IN TYPEOF(sdr.
related_shape_aspect.of_shape.definition))) )
= 1)) )) = 0)) ))
= 0);
wr5 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description IN ['outside boundary',
'complex boundary occurrence',
'partial circular boundary occurrence',
'closed circular boundary occurrence',
'open rectangular boundary occurrence',
'closed rectangular boundary occurrence']) ) = 1)) )) = 0))
OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,

```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND (('CAST_PARTS_SCHEMA.'
+ 'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (((
SIZEOF(['CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT'] *
TYPEOF(sdr.relating_shape_aspect)) = 1) AND (sdr.name =
'profile swept shape')) AND (sdr.relating_shape_aspect.
description = 'linear')) ) = 1) )) = 1))) = 0);
wr6 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description = 'complex boundary occurrence') )) = 1)) )) =
0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (SIZEOF([
'CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE',
'CAST_PARTS_SCHEMA.SQUARE_U_PROFILE',
'CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
'CAST_PARTS_SCHEMA.ROUNDED_U_PROFILE',
'CAST_PARTS_SCHEMA.VEE_PROFILE',
'CAST_PARTS_SCHEMA.TEE_PROFILE',
'CAST_PARTS_SCHEMA.LINEAR_PROFILE',
'CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
relating_shape_aspect)) = 1) )) = 1) )) = 1))) = 0);
wr7 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description = 'partial circular boundary occurrence') )) =
1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +

```



```

'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
TYPEOF(sdr.relatng_shape_aspect)) )) = 1) )) = 1)) )) = 0);
wr8 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description = 'closed circular boundary occurrence') )) = 1)) ))
= 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relatng_shape_aspect)) )) = 1) )) = 1)) )) = 0);
wr9 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description = 'open rectangular boundary occurrence') )) =
1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(
sdr.relatng_shape_aspect)) )) = 1) )) = 1)) )) = 0);
wr10: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description = 'closed rectangular boundary occurrence') ))
= 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,

```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
IN TYPEOF(sdr.relating_shape_aspect)) ) = 1) ) = 1)) ) =
0);
wr11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
description IN ['boundary occurrence',
'complex boundary occurrence',
'partial circular boundary occurrence',
'closed circular boundary occurrence',
'open rectangular boundary occurrence',
'closed rectangular boundary occurrence']) ) = 1)) ) = 0)
OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'removal direction')) ) = 1)) )
= 0);
wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit')) ) >=
0;
END_ENTITY; -- outside_profile
(*

```

## parabola

```

*)
ENTITY parabola
  SUBTYPE OF (conic);
  focal_dist : length_measure;
  WHERE
    wr1: focal_dist <> 0;
END_ENTITY; -- parabola
(*

```

## parallel\_offset

```

*)
ENTITY parallel_offset
  SUBTYPE OF (derived_shape_aspect);
  offset : measure_with_unit;
  WHERE
    wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- parallel_offset
(*

```

**parallelism\_tolerance**

```

*)
ENTITY parallelism_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) < 3;
  END_ENTITY; -- parallelism_tolerance

```

(\*

**parametric\_representation\_context**

```

*)
ENTITY parametric_representation_context
  SUBTYPE OF (representation_context);
  END_ENTITY; -- parametric_representation_context

```

(\*

**partial\_circular\_profile**

```

*)
ENTITY partial_circular_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
      .of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) >= 3)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
      'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'radius')) ) = 1)) ) = 0)) ) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
        it)) = 2) AND (it.name = 'sweep angle')) ) = 1)) ) = 0)) )
        = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELf,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
        pdr.used_representation)) AND (pdr.used_representation.name
        = 'profile limit')) ) <= 1)) ) = 0;
END_ENTITY; -- partial_circular_profile

```

(\*

## parting\_surface

\*)

```

ENTITY parting_surface
  SUBTYPE OF (shape_representation);
WHERE

wr1: (SIZEOF(QUERY( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'cope or drag') AND
  (it.description IN ['cope', 'drag']))=1);

wr2: (SIZEOF(QUERY( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'male or female') AND
  (it.description IN ['male', 'female']))=1);

wr3: (SIZEOF(QUERY( it <* SELF.items |
  (SIZEOF([
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *

```

```

        TYPEOF(it)) = 2) AND (it.name = 'offset distance'))>=1);

wr4:(SIZEOF(QUERY (rep <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (SIZEOF(['CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' ,
  'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION']
  * TYPEOF(rep.rep_2))>=1)))=1);

END_ENTITY; -- parting_surface
(*)
path
*)
ENTITY path
  SUPERTYPE OF (ONEOF (edge_loop,oriented_path))
  SUBTYPE OF (topological_representation_item);
  edge_list : LIST [1:?] OF UNIQUE oriented_edge;
  WHERE
    wr1: path_head_to_tail(SELF);
END_ENTITY; -- path
(*)
path_feature_component
*)
ENTITY path_feature_component
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
      SELF.of_shape.definition);
    wr2 : SELF.description IN ['partial circular','complete circular',
      'linear','complex'];
    wr3 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) = 1)) ) = 0);
    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
      'orientation')) ) = 1)) ) = 0)) ) = 0);
    wr5 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +

```

```

'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 3)) ) = 0)) ) = 0);
wr6 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'radius')) ) = 1)) ) = 0)) ) = 0);
wr7 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) ) = 1)) )
= 0)) ) = 0);
wr8 : (NOT (SELF.description = 'complete circular')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 2)) ) = 0)) ) = 0);
wr9 : (NOT (SELF.description = 'complete circular')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',

```

```

    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'radius')) ) = 1)) ) = 0)) ) = 0);
wr10: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(impl_rep.
    used_representation.items) = 2)) ) = 0)) ) = 0);
wr11: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'distance')) ) = 1)) ) = 0)) ) = 0);
wr12: (NOT (SELF.description = 'linear')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
    TYPEOF(pdr.used_representation)) ) = 1)) ) = 0);
wr13: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
    USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
    'CAST_PARTS_SCHEMA.PATH_SHAPE_REPRESENTATION' IN TYPEOF(pdr
    .used_representation)) AND (pdr.used_representation.name =
    'sweep path')) AND (SIZEOF(QUERY ( srwp_i <* pdr.
    used_representation.items | (srwp_i.name = 'profile shape') ) )
    = 1)) ) = 1)) ) = 0);
END_ENTITY; -- path_feature_component
(*)
path_shape_representation
*)
ENTITY path_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) >= 1;
    wr2: SIZEOF(QUERY ( i <* SELF.items | (SIZEOF([
      'CAST_PARTS_SCHEMA.BOUNDED_CURVE',
      'CAST_PARTS_SCHEMA.EDGE_CURVE', 'CAST_PARTS_SCHEMA.PATH'] *

```

```

        TYPEOF(i)) = 1) )) >= 1;
    END_ENTITY; -- path_shape_representation
    (*
pattern_equipment_available
    *)
    ENTITY pattern_equipment_available
        SUBTYPE OF (customer_casting_requirement);
    WHERE

    (* ----- core_box_cavity_id ----- *)

    wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'core box cavity id')
        ) ) =1 )))=0));

    (* ----- core_box_id ----- *)

    wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'core box id')
        ) ) =1 ))) =0));

    (* ----- number_of_core_boxes ----- *)

    wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component))) AND
        (it.name = 'number of core boxes')
        ) ) =1 )))=0));

    (* ----- number_of_cores_per_box ----- *)

    wr4: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component))) AND
        (it.name = 'number of core per box')
        ) ) =1 ))) =0));

    (* ----- number_of_cores_per_casting ----- *)

```



```

wr5:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of cores per casting'
  )) =1 ))) =0));

```

```
(* ----- number_of_patterns ----- *)
```

```

wr6:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component))) AND
  (it.name = 'number of patterns'
  )) =1 ))) =0));

```

```
(* ----- condition ----- *)
```

```

wr7:(NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'condition') AND
  (it.description IN ['good', 'bad', 'fair', 'new'] )
  ))) =1 )))=0));

```

```
(* ----- core_box_material ----- *)
```

```

wr8: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name = 'core box material') AND
  (SIZEOF(QUERY ( mfuo <*
  USEDIN(pdr.related_property_definition.definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (mfuo.name IN [
  'plaster',
  'expandable polystyrene',
  'iron',
  'soft wood',
  'shell',
  'aluminum',
  'epoxy',
  'wood metal reinforced',
  'brass',
  'hard_wood',
  'urethane',
  'other',

```

```
'steel',
'ceramic',
'polymethylmethacrylate',
'plywood']) )
= 1)) ) =0));
```

```
(* ----- pattern_impression_id ----- *)
```

```
wr9: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'pattern impression id')
))) =1 )) =0));
```

```
(* ----- pattern_material ----- *)
```

```
wr10: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
((pdr.name = 'pattern material') AND
(SIZEOF(QUERY ( mfuo <*
USEDIN(pdr.related_property_definition.definition,
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
(mfuo.name IN [
'plaster',
'expandable polystyrene',
'iron',
'soft wood',
'shell',
'aluminum',
'epoxy',
'wood metal reinforced',
'brass',
'hard wood',
'urethane',
'other',
'steel',
'ceramic',
'polymethylmethacrylate',
'plywood']) )
= 1)) ) =0));
```

```
(* ----- core_per_core_box_per_assembly ----- *)
```

```
wr11: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'core per core box per assembly') AND
(it.description IN ['same core','different core'] )
```

```

))) =1 )))=0));

(* ----- pattern ----- *)

wr12: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  (it.name = 'pattern') AND
  (it.description IN [ 'loose pattern with follow on board',
    'loose solid pattern',
    'loose split pattern', 'mounted cope and drag boards',
    'mounted cope and drag flaskless', 'mounted cope and drag inserts',
    'mounted snapboard', 'mounted match plate', 'sweep pattern'] )
  ))) =1 )))=0));

(* ----- to flask (as flask_size_for_mounted_patterns)----- *)

wr13: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flask size for mounted patterns') AND
  ('CAST_PARTS_SCHEMA.FLASK'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);

(* ----- to flask (as flask_size_for_plate_patterns)----- *)

wr14: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flask size for plate pattern') AND
  ('CAST_PARTS_SCHEMA.FLASK'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);

(* ----- to flaskless(as flaskless_pattern)----- *)

wr15: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ((pdr.name= 'flaskless pattern') AND
  ('CAST_PARTS_SCHEMA.FLASKLESS'
  IN TYPEOF (pdr.related_property_definition.definition)
  )) )) =1);

(* ***** to in_facility_location (as tooling_location) ***** *)
(* ***** to person_and_organization (as tooling_location) ***** *)

wr16: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```

```

('CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
 IN TYPEOF(pdr.used_representation)
))) =1) OR
(SIZEOF(QUERY(ada <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
(ada.role.name='tooling location')))=1);

```

```
END_ENTITY; -- pattern_equipment_available
```

```
(*
```

### **pattern\_offset\_membership**

```
*)
```

```

ENTITY pattern_offset_membership
  SUBTYPE OF (feature_component_relationship);
  WHERE
    wr1 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
      relating_shape_aspect,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | ((
      'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
      QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.
      related_shape_aspect.of_shape,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
      | (SIZEOF(['CAST_PARTS_SCHEMA.CIRCULAR_PATTERN',
      'CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(pdr.
      definition)) = 1) )) = 0) )) = 0;
    wr2 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
      related_shape_aspect,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((
      'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
      'modified pattern') AND (
      'CAST_PARTS_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
      relating_shape_aspect))) ) >= 1;
    wr3 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
      related_shape_aspect,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((
      'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
      'CAST_PARTS_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
      relating_shape_aspect)) AND (NOT (SIZEOF(QUERY ( modfcr <*
      QUERY ( modsar <* USEDIN(fcr.relatng_shape_aspect,
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
      SIZEOF(['CAST_PARTS_SCHEMA.CIRCULAR_PATTERN',
      'CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(modsar.
      related_shape_aspect.of_shape.definition)) = 1) AND (modsar
      :<>: fcr)) ) | (NOT (modfcr.related_shape_aspect.of_shape.
      definition :=: SELF.relatng_shape_aspect.of_shape.

```

```

        definition)) )) = 0))) )) = 0;
wr4 : (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 2)) ))
    = 0);
wr5 : (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
    = 0);
wr6 : (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
        SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
wr7 : (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
        it)) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
        measure_with_unit.value_component))) AND (it.name =
        'index_number')) )) = 1)) )) = 0)) )) = 0);
wr8 : (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT' IN
        TYPEOF(it)) AND (it.name = 'offset')) )) = 1)) )) = 0)) ))
    = 0);
wr9 : (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(

```

```

QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 3)) ) = 0)) ) = 0);
wr10: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
relating_shape_aspect.of_shape.definition)) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
it)) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component))) AND (it.name =
'row index')) ) = 1)) ) = 0)) ) = 0);
wr11: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
relating_shape_aspect.of_shape.definition)) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
it)) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component))) AND (it.name =
'column index')) ) = 1)) ) = 0)) ) = 0);
wr12: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
relating_shape_aspect.of_shape.definition)) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(it))
AND (it.name = 'offset distance')) ) = 1)) ) = 0)) ) = 0);
wr13: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.

```

```

relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'offset direction')) )) = 1)) ))
= 0);
END_ENTITY; -- pattern_offset_membership
(*)
pattern_omit_membership
*)
ENTITY pattern_omit_membership
SUBTYPE OF (feature_component_relationship);
WHERE
wr1: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
relating_shape_aspect,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | ((
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.related_shape_aspect
.of_shape, 'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
| ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(
pd)) ) | (SIZEOF(['CAST_PARTS_SCHEMA.CIRCULAR_PATTERN',
'CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(pdr.
definition)) = 1) )) = 0) )) = 0;
wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
related_shape_aspect,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
'modified pattern') AND (
'CAST_PARTS_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
relating_shape_aspect))) )) >= 1;
wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
related_shape_aspect,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
'CAST_PARTS_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
relating_shape_aspect)) AND (NOT (SIZEOF(QUERY ( modfcr <*
QUERY ( modsar <* USEDIN(fcr.related_shape_aspect,
'CAST_PARTS_SCHEMA.' +
'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
SIZEOF(['CAST_PARTS_SCHEMA.CIRCULAR_PATTERN',
'CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(modsar.
related_shape_aspect.of_shape.definition)) = 1) AND (modsar
:<>: fcr)) ) | (NOT (modfcr.related_shape_aspect.of_shape.
definition ::= SELF.related_shape_aspect.of_shape.

```

```

        definition)) )) = 0))) )) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) )) =
    0;
wr5: (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
    SIZEOF(pdr.used_representation.items) = 1)) )) = 0)) )) = 0);
wr6: (NOT ('CAST_PARTS_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | (((
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
    measure_with_unit.value_component))) AND (it.name =
    'index number')) )) = 1)) )) = 0)) )) = 0);
wr7: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
    SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
wr8: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | (((
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
    AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
    measure_with_unit.value_component))) AND (it.name =
    'row index')) )) = 1)) )) = 0)) )) = 0);
wr9: (NOT ('CAST_PARTS_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(SELF.
    relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF.related_shape_aspect,

```



```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
measure_with_unit.value_component))) AND (it.name =
'column index')) ) = 1)) ) = 0)) ) = 0);
END_ENTITY; -- pattern_omit_membership

```

(\*

## pattern\_plate

\*)

```

ENTITY pattern_plate
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
  (* ----- type_of_plate ----- *)

wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
  (co.name IN ['wood','metal','metal insert','wood insert'])) )
  = 1;

(* ----- pattern_plate to item_size (as pattern_plate_dimensions) --- *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='pattern plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
  (sar.description = 'pattern plate usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))
  )) =1 ))) >=1;

(* - pattern_plate to location_element (as mounting_hole_locations)---- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
  TYPEOF(pdr.used_representation)) AND
  (pdr.used_representation.name='location')))=1;

(* ----- pattern_plate to flask (as defines_flask_type) ----- *)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='pattern plate occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND

```

```

(sar.description = 'flask reference usage') AND
('CAST_PARTS_SCHEMA.FLASK' IN TYPEOF(sar.relater_shape_aspect))
)) =1 ))) >=1;

(* ----- pattern_plate to gating_system (as gate_rigging) ----- *)

wr5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='pattern plate occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'gating system reference usage') AND
('CAST_PARTS_SCHEMA.GATING_SYSTEM'
IN TYPEOF(sar.relater_shape_aspect))
)) =1 ))) >=1;

(* ----- pattern_plate to mould_locks (as lock_definitions) ----- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='pattern plate occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'mould lock reference usage') AND
('CAST_PARTS_SCHEMA.MOULD_LOCK' IN TYPEOF(sar.relater_shape_aspect))
)) <=1 ))) >=1;

(* ----- pattern_plate to parting_surface (as parting_line) ---- *)

wr7: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
(('CAST_PARTS_SCHEMA.PARTING_SURFACE' IN
TYPEOF(pdr.used_representation)) AND
(pdr.used_representation.name='parting line')))=1;

(* ----- pattern_plate to pin_center (as pin_center_definition) -- *)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='pattern plate occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'pin center reference usage') AND
('CAST_PARTS_SCHEMA.PIN_CENTER' IN TYPEOF(sar.relater_shape_aspect))
)) <=1 ))) >=1;

(* ----- pattern_plate to moulding_equipment (as moulding_machine)---- *)

wr9: (SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
(('CAST_PARTS_SCHEMA.ACTION_METHOD'
IN TYPEOF(adr.assigned_action_method)) AND
(SIZEOF(QUERY(ar <* USEDIN(adr.assigned_action_method,

```

```

        'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
        ('CAST_PARTS_SCHEMA.MOULDING_EQUIPMENT' IN TYPEOF(ar) ))=1)))=1);
END_ENTITY; -- pattern_plate
(*)
payment_and_shipping
*)
    ENTITY payment_and_shipping
        SUBTYPE OF (customer_casting_requirement);
        WHERE
    (* ----- customer_payment ----- *)

    wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'customer_payment')
        ) )) =1 )))=0));

    (* ----- packaging_or_creating ----- *)

    wr2: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'packaging or creating')
        ) )) =1 )))=0));

    (* ----- ship_casting ----- *)

    wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        ( SIZEOF(QUERY ( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND
        (it.name = 'ship_casting')
        ) )) =1 )))=0));

    END_ENTITY; -- payment_and_shipping
    (*)
pcurve
*)
    ENTITY pcurve
        SUBTYPE OF (curve);
        basis_surface      : surface;
        reference_to_curve : definitional_representation;
        WHERE
            wr1: SIZEOF(reference_to_curve\representation.items) = 1;
            wr2: 'CAST_PARTS_SCHEMA.CURVE' IN TYPEOF(reference_to_curve\
                representation.items[1]);
            wr3: reference_to_curve\representation.items[1]\
                geometric_representation_item.dim = 2;

```

```

    END_ENTITY; -- pcurve
    (*
permanent_moulding_process
    *)
    ENTITY permanent_moulding_process
        SUBTYPE OF (casting_activity);
        WHERE
    (* ----- moulding_process ----- *)

        wr1: (SELF.description IN ['static','tilt','low pressure']);

    (* ----- number_of_impressions ----- *)

wr2:    SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
        (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
            TYPEOF(pdr.representation)) AND
            (SIZEOF(QUERY (it <* pdr.representation.items |
                (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
                ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
                    IN TYPEOF(it\measure_with_unit.value_component)) AND
                (it.name = 'number of impressions')) ) =1 ))))=1;

    (* ---- to die_mould_vent (as ejection_definition) ----- *)

wr3: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
            IN TYPEOF(ama)) AND
            (SIZEOF(QUERY(eds <* ama.items |
                (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
                ('CAST_PARTS_SCHEMA.DIE_MOULD_VENT' IN TYPEOF(eds.definition))
                )) =1 ))))=0));

    (* ---- to ejector_system (as vent_definition) ----- *)

wr4: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
            IN TYPEOF(ama)) AND
            (SIZEOF(QUERY(eds <* ama.items |
                (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
                ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM' IN TYPEOF(eds.definition))
                )) =1 ))))=0));

    (* ---- (as permanent_die_definition) ----- *)

wr5: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
            IN TYPEOF(ama)) AND
            (SIZEOF(QUERY(eds <* ama.items |
                (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
                ('CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD' IN TYPEOF(eds.definition))

```

```

    ))) =1 )))=0));

(* -- permanent_mould_process to ingate (as metal_flow_system) ----- *)

wr6:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.INGATE' IN TYPEOF(eds.definition))
  ))) =1 )))=0));

(* ---- p to runner (as metal_flow_system) ----- *)

wr7:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(eds.definition))
  ))) =1 )))=0));

(* ---- to sprue (as metal_flow_system) ----- *)

wr8:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(eds.definition))
  ))) =1 )))=0));

```

```
END_ENTITY; -- permanent_moulding_process
```

```
(*
```

## **perpendicular\_to**

```
*)
```

```
ENTITY perpendicular_to
  SUBTYPE OF (derived_shape_aspect);
  WHERE
    wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- perpendicular_to
```

```
(*
```

## **perpendicularity\_tolerance**

```
*)
```

```
ENTITY perpendicularity_tolerance
  SUBTYPE OF (geometric_tolerance_with_datum_reference);
  WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
      datum_system) <= 3;
END_ENTITY; -- perpendicularity_tolerance
```

(\*

### **person**

\*)

```
ENTITY person;
    id          : identifier;
    last_name   : OPTIONAL label;
    first_name  : OPTIONAL label;
    middle_names : OPTIONAL LIST [1:?] OF label;
    prefix_titles : OPTIONAL LIST [1:?] OF label;
    suffix_titles : OPTIONAL LIST [1:?] OF label;
    WHERE
        wr1: EXISTS(last_name) OR EXISTS(first_name);
END_ENTITY; -- person
```

(\*

### **person\_and\_organization**

\*)

```
ENTITY person_and_organization;
    the_person      : person;
    the_organization : organization;
    DERIVE
        name          : label := get_name_value(SELF);
        description   : text := get_description_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
        wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization
```

(\*

### **person\_and\_organization\_assignment**

\*)

```
ENTITY person_and_organization_assignment
    ABSTRACT SUPERTYPE;
    assigned_person_and_organization : person_and_organization;
    role                             : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment
```

(\*

### **person\_and\_organization\_role**

\*)

```
ENTITY person_and_organization_role;

    name : label;
    DERIVE
        description : text := get_description_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization_role
```

(\*

### **personal\_address**

\*)

```

ENTITY personal_address
  SUBTYPE OF (address);
  people      : SET [1:?] OF person;
  description : OPTIONAL text;
END_ENTITY; -- personal_address
(*)
pin_center
*)
ENTITY pin_center
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
    wr1: SIZEOF(QUERY ( pdr <*
      get_property_definition_representations(SELF) |
      (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
          ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
            'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
            * TYPEOF(it)) = 2) AND
            (it.name = 'dimension')) = 1 )))=1;

    END_ENTITY; -- pin_center
(*)
pin_tip
*)
ENTITY pin_tip
  SUBTYPE OF (ejector_design_feature);
  WHERE
    wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
      ((sa.description='pin tip occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
          'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
          (sar.description = 'pin tip shape usage') AND
          ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
          )) = 1 )))) = 1;
    END_ENTITY; -- pin_tip
(*)
placed_datum_target_feature
*)
ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
  WHERE
    wr1 : SELF.description IN ['point','line','rectangle','circle'];
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) = 1)) = 0;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) )) = 1)) )) = 0)) )) = 0;
wr4 : (NOT (SELF.description = 'point')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(dtm_rep.
used_representation.items) = 1)) )) = 0)) )) = 0);
wr5 : (NOT (SELF.description IN ['line','circle'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(dtm_rep.
used_representation.items) = 2)) )) = 0)) )) = 0);
wr6 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(dtm_rep.
used_representation.items) = 3)) )) = 0)) )) = 0);
wr7 : (NOT (SELF.description = 'circle')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'target diameter')) )) = 1)) )) = 0)) ))

```



```

= 0);
wr8 : (NOT (SELF.description = 'line')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'target length')) ) = 1)) ) = 0)) )
= 0);
wr9 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'target length')) ) = 1)) ) = 0)) )
= 0);
wr10: (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
dtm_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'target width')) ) = 1)) ) = 0)) ) =
0);
END_ENTITY; -- placed_datum_target_feature
(*)
placement
*)
ENTITY placement
  SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
    axis2_placement_3d))
  SUBTYPE OF (geometric_representation_item);
  location : cartesian_point;
END_ENTITY; -- placement

```

(\*

### **planar\_shape\_representation**

\*)

```
ENTITY planar_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ('CAST_PARTS_SCHEMA.PLANE' IN
      TYPEOF(it)) )) = 1;
  END_ENTITY; -- planar_shape_representation
```

(\*

### **plane**

\*)

```
ENTITY plane
  SUBTYPE OF (elementary_surface);
  END_ENTITY; -- plane
```

(\*

### **plane\_angle\_measure\_with\_unit**

\*)

```
ENTITY plane_angle_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.PLANE_ANGLE_UNIT' IN TYPEOF(SELF\
      measure_with_unit.unit_component);
  END_ENTITY; -- plane_angle_measure_with_unit
```

(\*

### **plane\_angle\_unit**

\*)

```
ENTITY plane_angle_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
      \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
      named_unit.dimensions.time_exponent = 0)) AND (SELF\
      named_unit.dimensions.electric_current_exponent = 0)) AND (
      SELF\named_unit.dimensions.
      thermodynamic_temperature_exponent = 0)) AND (SELF\
      named_unit.dimensions.amount_of_substance_exponent = 0)) AND
      (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
  END_ENTITY; -- plane_angle_unit
```

(\*

### **plus\_minus\_tolerance**

\*)

```
ENTITY plus_minus_tolerance;
  range : tolerance_method_definition;
  toleranced_dimension : dimensional_characteristic;
  UNIQUE
    url : toleranced_dimension;
  END_ENTITY; -- plus_minus_tolerance
```

(\*

### **pocket**

```

*)
ENTITY pocket
  SUBTYPE OF (feature_definition);
  WHERE
    wr1 : SELF\characterized_object.description IN ['closed rectangular',
      'open rectangular','complex','circular cutout',
      'complex cutout','recess'];
    wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd) ) )
      | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'pocket depth occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND (sar.name =
      'pocket depth')) AND (
      'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
      sar))) ) | (('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
      TYPEOF(sdr.relateing_shape_aspect)) AND (sdr.
      relating_shape_aspect.description = 'linear')) ) = 1)) )
      = 1)) ) = 0;
    wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | (
      'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
      TYPEOF(pdr.used_representation) ) ) = 1;
    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) AND ((1 <= SIZEOF(pdr.
      used_representation.items)) AND (SIZEOF(pdr.
      used_representation.items) <= 2))) ) = 1) ) = 1;
    wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
      used_representation.items | ((srwp_i.name = 'orientation')
      OR (srwp_i.name = 'fillet radius')) ) = SIZEOF(pdr.
      used_representation.items))) ) = 1) ) = 1;
    wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'fillet radius')) ) <= 1)) ) = 0)) )
= 0;
wr7 : (NOT (SELF\characterized_object.description IN ['complex',
'non-circular cutout','recess'])) OR (SIZEOF(
QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (SIZEOF([
'CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE',
'CAST_PARTS_SCHEMA.SQUARE_U_PROFILE',
'CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
'CAST_PARTS_SCHEMA.ROUNDED_U_PROFILE',
'CAST_PARTS_SCHEMA.VEE_PROFILE',
'CAST_PARTS_SCHEMA.TEE_PROFILE',
'CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
relating_shape_aspect)) = 1)) = 1)) = 1)) = 0);
wr8 : (NOT (SELF\characterized_object.description =
'closed rectangular')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'closed boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE'
IN TYPEOF(sdr.relying_shape_aspect)) ) = 1)) ) = 1)) )
= 0);
wr9 : (NOT (SELF\characterized_object.description =
'open rectangular')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,

```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'open boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(
sdr.relater_shape_aspect))) = 1)) = 1)) = 0);
wr10: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'bottom condition occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'pocket bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.POCKET_BOTTOM' IN
TYPEOF(sdr.relater_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.POCKET' IN TYPEOF(sdr.
relater_shape_aspect.of_shape.definition))) = 1)) = 1)) = 0;
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'change in boundary occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | ((sar.description =
'taper usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | ((('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(
fcr.relater_shape_aspect)) AND ('CAST_PARTS_SCHEMA.POCKET'
IN TYPEOF(fcr.relater_shape_aspect.of_shape.definition)))
AND (fcr.relater_shape_aspect.description IN ['angle taper',
'directed taper']))) = 1)) = 1)) = 0;
wr12: (NOT (SELF\characterized_object.description =
'circular cutout')) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'enclosed boundary occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (

```

```

'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relating_shape_aspect)) ) = 1)) ) = 1)) ) = 0);
wr13: (NOT (SELF\characterized_object.description IN [
'circular cutout','complex cutout'])) OR (SIZEOF(
QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'bottom condition occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'pocket bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | ((('CAST_PARTS_SCHEMA.POCKET_BOTTOM' IN
TYPEOF(sdr.relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.POCKET' IN TYPEOF(sdr.
related_shape_aspect.of_shape.definition))) AND (sdr.
relating_shape_aspect.description = 'through')) ) = 1)) )
= 1)) ) = 0);
wr14: (NOT (SELF\characterized_object.description = 'recess')) OR (
SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'bottom condition occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'pocket bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | ((('CAST_PARTS_SCHEMA.POCKET_BOTTOM' IN
TYPEOF(sdr.relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.POCKET' IN TYPEOF(sdr.
related_shape_aspect.of_shape.definition))) AND (sdr.
relating_shape_aspect.description IN ['planar','complex'])) )
= 1)) ) = 1)) ) = 0);
wr15: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
IN TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit')) ) ) >=
0;
wr16: (NOT (SELF\characterized_object.description IN [
'closed rectangular','open rectangular','complex','recess']))
OR (SIZEOF(QUERY ( pds <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))

```

```

AND (csa.name = 'uncut area')) AND (SIZEOF(QUERY ( sar <*
csa.component_relationships | ((
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) AND (SIZEOF(['CAST_PARTS_SCHEMA.BOSS',
'CAST_PARTS_SCHEMA.PROTRUSION'] * TYPEOF(sar.
related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1);
END_ENTITY; -- pocket
(*)
pocket_bottom
*)
ENTITY pocket_bottom
SUBTYPE OF (shape_aspect);
WHERE
wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
SELF.of_shape.definition);
wr2 : SELF.description IN ['planar','complex','through'];
wr3 : (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'floor normal')) )) = 1)) )) = 0);
wr4 : (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'floor location')) )) = 1)) )) =
0);
wr5 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr
.used_representation)) AND (pdr.used_representation.name =
'floor face')) )) = 1)) )) = 0);
wr6 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) )) = 1)) )) = 0);
wr7 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) <= 1)) ) = 0)) ) = 0);
wr8 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) = 0)) ) = 0);
wr9 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'radius')) ) <= 1)) ) = 0)) ) = 0);
wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') | ((sar.description =
        'pocket bottom usage') AND (
        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
        'bottom condition occurrence') AND (
        'CAST_PARTS_SCHEMA.POCKET' IN TYPEOF(fcr.
        related_shape_aspect.of_shape.definition))) AND (
        'CAST_PARTS_SCHEMA.POCKET_BOTTOM' IN TYPEOF(fcr.
        relating_shape_aspect))) ) >= 1;
wr11: (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
        QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') | ((sar.description =
        'pocket bottom usage') AND (
        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | ((fcr.related_shape_aspect.description =
        'bottom condition occurrence') AND (fcr.
        related_shape_aspect.name IN ['pocket depth start',
        'pocket depth end']))) ) = 0);
END_ENTITY; -- pocket_bottom

```

(\*

## point

\*)

ENTITY point



```

    SUPERTYPE OF (cartesian_point)
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point
(*)
poly_loop
*)
    ENTITY poly_loop
        SUBTYPE OF (loop, geometric_representation_item);
        polygon : LIST [3:?] OF UNIQUE cartesian_point;
    END_ENTITY; -- poly_loop
(*)
polyline
*)
    ENTITY polyline
        SUBTYPE OF (bounded_curve);
        points : LIST [2:?] OF cartesian_point;
    END_ENTITY; -- polyline
(*)
position_tolerance
*)
    ENTITY position_tolerance
        SUBTYPE OF (geometric_tolerance);
    WHERE
    WR1: ( NOT ('CAST_PARTS_SCHEMA.' +
'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE' IN TYPEOF (SELF)))
    OR
    (SIZEOF (SELF\geometric_tolerance_with_datum_reference.
datum_system) <= 3);
    END_ENTITY; -- position_tolerance
(*)
pre_defined_item
*)
    ENTITY pre_defined_item;
        name : label;
    END_ENTITY; -- pre_defined_item
(*)
precision_qualifier
*)
    ENTITY precision_qualifier;
        precision_value : INTEGER;
    END_ENTITY; -- precision_qualifier
(*)
process_parameter_record
*)
    ENTITY process_parameter_record
        SUBTYPE OF (activity_result_record);
    WHERE
    (* - to property_parameter (as actual_parameter) ----- *)

    wr1: (SIZEOF(QUERY( it <* SELF.items |
        (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',

```

```

    'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
    * TYPEOF(it)) = 1) AND
    (it.name = 'actual parameter'))=1);

(* -- to property_parameter (as planned_parameter) ----- *)

wr2: (SIZEOF(QUERY( it <* SELF.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
    * TYPEOF(it)) = 1) AND
    (it.name = 'planned parameter'))=1);

END_ENTITY; -- process_parameter_record
(*)
process_plan_activity
*)
ENTITY process_plan_activity
    SUBTYPE OF (action_method);
    WHERE
    (* ----- activity_number ----- *)

wr1:  SIZEOF(QUERY ( pdr <* get_action_property_representation(SELF) |
    (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
    TYPEOF(pdr.representation)) AND
    (SIZEOF(QUERY (it <* pdr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'activity number')) ) =1 ))))=1;

(* ----- frequency ----- *)

wr2:  (SIZEOF(QUERY(ar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE' ) |
    (ar.name='frequency' ) ) <=1);

(* ---- to activity_execution_result to activity (as execution_result) - *)

wr3:  (SIZEOF(QUERY(ar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD') |
    ('CAST_PARTS_SCHEMA.ACTIVITY_EXECUTION_RESULT'
    IN TYPEOF(ar) ) ) <=1);

(* ----to organization (as performed_by_organization_id) ----- *)

wr4: (SIZEOF(QUERY ( aam <* USEDIN(SELF ,
    'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
    ((aam\organization_assignment.role.name='activity organization id') AND
    ('CAST_PARTS_SCHEMA.ORGANIZATION'
    IN TYPEOF(aam\organization_assignment.assigned_organization))))<=1);

```

```
(* ----- to special_instruction (as activity_information) ----- *)

wr5:      SIZEOF(QUERY ( pdr <* get_action_property_representation(SELf) |
  (('CAST_PARTS_SCHEMA.REPRESENTATION' IN
  TYPEOF(pdr.representation)) AND
  (SIZEOF(QUERY (it <* pdr.representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'special instruction')
  )))) >=0 ))))>=1;

(* --- to supplemental_document (as process_specification) -----*)

wr6:(NOT (('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELf))) OR
  (SIZEOF(QUERY ( adr <*
  SELf\action_method_with_associated_documents.documents |
  (('CAST_PARTS_SCHEMA.PROCESS_PLAN_SPECIFICATION'
  IN TYPEOF(adr)) AND
  (adr.kind.product_data_type = 'supplemental documents')))) >= 0);

(* ----- activity to performance_rate (as duration) ----- *)

wr7:      (NOT(SIZEOF(QUERY(ap <* USEDIN(SELf,
  'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.ALLOWED_TIME',
    'CAST_PARTS_SCHEMA.PRODUCTION_RATE'] * TYPEOF(ap)) >= 0)))=0));

(* --activity to design_reference (as graphics_representation) -----*)
(* --activity to view_reference (as graphics_representation) -----*)
(* --activity to illustration (as graphics_representation) -----*)

wr8: (NOT('CAST_PARTS_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
  IN TYPEOF(SELf)) OR
  (NOT (SIZEOF(QUERY( doc <* SELf.documents |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.DESIGN_REFERENCE',
    'CAST_PARTS_SCHEMA.VIEW_REFERENCE',
    'CAST_PARTS_SCHEMA.ILLUSTRATION'] * TYPEOF(doc)) >=0 ))) =0 )));

(* ----- activity to cast_part (as output_item) ----- *)

wr9: (SIZEOF(QUERY ( apr <* USEDIN(SELf ,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(apr)) AND
  (NOT(SIZEOF(QUERY(it <* apr.items |
  (SIZEOF([
    'CAST_PARTS_SCHEMA.CAST_PART',
    'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING'] * TYPEOF(it)) <=1 )
  )) =0 )) )) <=1);
```

## ISO 10303-223:2008 (E)

```
(* activity to generic_manufacturing_resource (as consumable_resource) *)
```

```
wr10: (SIZEOF(QUERY(ar <* USEDIN(SELf,  
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE' ) |  
  ('CAST_PARTS_SCHEMA.GENERIC_MANUFACTURING_RESOURCE'  
  IN TYPEOF(ar))))>=0);
```

```
END_ENTITY; -- process_plan_activity
```

```
(*
```

### **process\_plan\_security**

```
*)
```

```
ENTITY process_plan_security
```

```
  SUBTYPE OF (security_classification);
```

```
  WHERE
```

```
    wr1: SIZEOF(QUERY ( da <* USEDIN(SELf,  
      'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') | (((  
      'CAST_PARTS_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da)) AND (  
      'CAST_PARTS_SCHEMA.CALENDAR_DATE'  
      IN TYPEOF(da.assigned_date)))  
      AND (da.role.name = 'classification date')) ) ) = 1;
```

```
    wr2: SIZEOF(QUERY ( da <* USEDIN(SELf,  
      'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') | (((  
      'CAST_PARTS_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da)) AND (  
      'CAST_PARTS_SCHEMA.CALENDAR_DATE'  
      IN TYPEOF(da.assigned_date)))  
      AND (da.role.name = 'declassification date')) ) ) = 1;
```

```
    wr3: SIZEOF(QUERY ( ap <* USEDIN(SELf,  
      'CAST_PARTS_SCHEMA.SECURITY_CLASSIFICATION_ASSIGNMENT.' +  
      'ASSIGNED_SECURITY_CLASSIFICATION') |  
      (('CAST_PARTS_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'  
      IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (  
      'CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(it)) ) )  
      = 1))) ) ) = 0;
```

```
    wr4: SIZEOF(QUERY ( ap <* USEDIN(SELf, 'CAST_PARTS_SCHEMA.' +  
      'SECURITY_CLASSIFICATION_ASSIGNMENT.ASSIGNED_SECURITY_CLASSIFICATION')  
      | (('CAST_PARTS_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'  
      IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (  
      'CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(it)) ) ) =  
      1))) ) ) = 0;
```

```
END_ENTITY; -- process_plan_security
```

```
(*
```

### **process\_plan\_specification**

```
*)
```

```
ENTITY process_plan_specification
```

```
  SUBTYPE OF (document);
```

```
  WHERE
```

```
    wr1: SIZEOF(QUERY ( duc <* USEDIN(SELf,  
      'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc  
      .subject_element = 'revision')) ) = 1;
```

```
    wr2: SIZEOF(QUERY ( duc <* USEDIN(SELf,  
      'CAST_PARTS_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') | (duc  
      .subject_element = 'subclass')) ) = 1;
```

```
END_ENTITY; -- process_plan_specification
```

```

(*)
process_plan_version
*)
  ENTITY process_plan_version
    SUBTYPE OF (product_definition_process);
  WHERE

  wr1: (NOT(SIZEOF(QUERY ( ppp <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.PROCESS') |
    (SIZEOF(['CAST_PARTS_SCHEMA.DESIGN_PART',
      'CAST_PARTS_SCHEMA.CAST_PART',
      'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
      'CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION']
    * TYPEOF(ppp.defined_product)
      )=1) ))=0));

  wr2: (SIZEOF(QUERY(act <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATING_ACTION') |
    ('CAST_PARTS_SCHEMA.ACTIVITY_EXECUTION_RESULT'
    IN TYPEOF(act.related_action)
      )=1));

  (* process_plan to casting_activity (as activities) *)

  wr3: (SIZEOF(QUERY ( duc <* USEDIN(SELF.chosen_method,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    ('CAST_PARTS_SCHEMA.CASTING_ACTIVITY' IN TYPEOF(duc.related_method)
      ))=1));

  (* to manufacturing_process (as activities to produce part) *)

  wr4: (NOT(SELF\action.id = 'mini plan')) AND
    (NOT(SIZEOF(QUERY ( duc <* USEDIN(SELF.chosen_method,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP' IN TYPEOF(duc)) AND
    ( SIZEOF(['
    'CAST_PARTS_SCHEMA.NON_MACHINING_PROCESS',
    'CAST_PARTS_SCHEMA.MACHINING_PROCESS']
    * TYPEOF(duc.related_method)=1))))=0));

  (* ----- to material_usage as amount of raw materials) ----- *)

  wr5: (NOT(SELF\action.id = 'mini plan')) AND
    (NOT(SIZEOF(QUERY( act <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_ASSIGNMENT' IN TYPEOF(act)) AND
    (SIZEOF(QUERY( mu <* act.items |
    'CAST_PARTS_SCHEMA.MATERIAL_USAGE' IN TYPEOF(mu))=1))))=0));

  (* ----- to ordered_part (quantity_of_parts) ----- *)

  wr6: (NOT(SELF\action.id = 'mini plan')) AND
    (NOT (SIZEOF(QUERY( act <* USEDIN(SELF,

```

```
'CAST_PARTS_SCHEMA.ORDERED_PART.ITEMS') |
(SIZEOF(QUERY ( pd <* USEDIN(act,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'quantity'))
)) =1 )) )) =0 )) )) =0 ))) =0));
```

(\* to property\_parameter (as process plan properties) \*)

```
wr7:(NOT(SELF\action.id = 'mini plan')) AND
(NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY ( it <* apr.representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM']
* TYPEOF (it) )=1 ) AND
(it.name='process plan properties' )
))=1)))=0))))=0));
```

(\* to special\_instruction (as process plan information) \*)

```
wr8: (NOT(SELF\action.id = 'mini plan')) AND
(NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY ( it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'special instruction'))
)) = 1))) = 0)))) = 0));
```

END\_ENTITY; -- process\_plan\_version

(\*

### **process\_product\_association**

\*)

```
ENTITY process_product_association;
    name          : label;
    description    : text;
    defined_product : characterized_product_definition;
    process        : product_definition_process;
END_ENTITY; -- process_product_association
```

(\*

### **process\_property\_association**

\*)

```
ENTITY process_property_association;
    name          : label;
    description    : text;
    process        : property_process;
```

```

        property_or_shape : property_or_shape_select;
    END_ENTITY; -- process_property_association
(*)
process_requirement
*)
    ENTITY process_requirement
        SUBTYPE OF (customer_casting_requirement);
        WHERE
            wr1: NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
                | (SIZEOF(QUERY ( it <* pdr.used_representation.items | ((
                'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
                TYPEOF(it)) AND (it.name = 'process name')) )) = 1) )) = 0);
    END_ENTITY; -- process_requirement
(*)
product
*)
    ENTITY product;
        id : identifier;
        name : label;
        description : OPTIONAL text;
        frame_of_reference : SET [1:?] OF product_context;
    END_ENTITY; -- product
(*)
product_category
*)
    ENTITY product_category;
        name : label;
        description : OPTIONAL text;
        DERIVE
            id : identifier := get_id_value(SELF);
        WHERE
            wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
                'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
    END_ENTITY; -- product_category
(*)
product_context
*)
    ENTITY product_context
        SUBTYPE OF (application_context_element);
        discipline_type : label;
    END_ENTITY; -- product_context
(*)
product_definition
*)
    ENTITY product_definition;
        id : identifier;
        description : OPTIONAL text;
        formation : product_definition_formation;
        frame_of_reference : product_definition_context;
        DERIVE

```

```

        name : label := get_name_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
    END_ENTITY; -- product_definition
(*)
product_definition_context
*)
    ENTITY product_definition_context
        SUBTYPE OF (application_context_element);
        life_cycle_stage : label;
    END_ENTITY; -- product_definition_context
(*)
product_definition_effectivity
*)
    ENTITY product_definition_effectivity
        SUBTYPE OF (effectivity);
        usage : product_definition_relationship;
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) = 0;
    END_ENTITY; -- product_definition_effectivity
(*)
product_definition_formation
*)
    ENTITY product_definition_formation;
        id : identifier;
        description : OPTIONAL text;
        of_product : product;
    UNIQUE
        url : id, of_product;
    END_ENTITY; -- product_definition_formation
(*)
product_definition_formation_relationship
*)
    ENTITY product_definition_formation_relationship;
        id : identifier;
        name : label;
        description : OPTIONAL text;
        relating_product_definition_formation : product_definition_formation;
        related_product_definition_formation : product_definition_formation;
    END_ENTITY; -- product_definition_formation_relationship
(*)
product_definition_process
*)
    ENTITY product_definition_process
        SUBTYPE OF (action);
        identification : identifier;
    INVERSE
        product_definitions : SET [1:?] OF process_product_association FOR
            process;

```



```
END_ENTITY; -- product_definition_process
(*)
```

### **product\_definition\_relationship**

```
*)
ENTITY product_definition_relationship;
    id                : identifier;
    name              : label;
    description       : OPTIONAL text;
    relating_product_definition : product_definition;
    related_product_definition  : product_definition;
END_ENTITY; -- product_definition_relationship
```

```
(*)
```

### **product\_definition\_shape**

```
*)
ENTITY product_definition_shape
    SUBTYPE OF (property_definition);
    UNIQUE
    url : definition;
    WHERE
        wr1: SIZEOF(['CAST_PARTS_SCHEMA.CHARACTERIZED_PRODUCT_DEFINITION',
                    'CAST_PARTS_SCHEMA.CHARACTERIZED_OBJECT'] * TYPEOF(SELF\
                    property_definition.definition)) > 0;
END_ENTITY; -- product_definition_shape
```

```
(*)
```

### **product\_definition\_usage**

```
*)
ENTITY product_definition_usage
    SUPERTYPE OF (ONEOF (make_from_usage_option, assembly_component_usage))
    SUBTYPE OF (product_definition_relationship);
    UNIQUE
    url : id, relating_product_definition, related_product_definition;
    WHERE
        wr1: acyclic_product_definition_relationship(SELF, [SELF\
                product_definition_relationship.related_product_definition],
                'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_USAGE');
END_ENTITY; -- product_definition_usage
```

```
(*)
```

### **product\_definition\_with\_associated\_documents**

```
*)
ENTITY product_definition_with_associated_documents
    SUBTYPE OF (product_definition);
    documentation_ids : SET [1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents
```

```
(*)
```

### **product\_material\_composition\_relationship**

```
*)
ENTITY product_material_composition_relationship
    SUBTYPE OF (product_definition_relationship);
    class                : label;
    constituent_amount   : SET [1:?] OF measure_with_unit;
    composition_basis    : label;
```

```

        determination_method : text;
    END_ENTITY;
    (*
product_related_product_category
    *)
    ENTITY product_related_product_category
        SUBTYPE OF (product_category);
        products : SET [1:?] OF product;
    END_ENTITY; -- product_related_product_category
    (*
production_core_box
    *)
    ENTITY production_core_box
        SUBTYPE OF (sand_cast_design_feature);
        WHERE

    (* ***** box_type ***** *)

    wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'core box type')))) =1 ))))=1;

    (* ***** vent_type ***** *)

    wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'core vent type')))) =1 ))))=1;

    (* ----- core_box_material ----- *)

    WR3: SIZEOF(QUERY ( co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
        (co.name IN ['wood','cast iron','urethane','aluminum'])) ) ) = 1;

    (* ----- number_of_impressions ----- *)

    wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'number of impressions')) ) ) =1 ))))=1;

    (* -----shrink_factor----- *)

```

```

wr5:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor')) ) =1 ))))=1;

(* -----production_core_box to casting_insert (as inserts) --- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'casting insert reference usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_INSERT'
  IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

(* ---production_core_box to chaplet_pad (as chaplet_pad_definition) --- *)

wr7:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chaplet pad reference usage') AND
  ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

(* -----production_core_box to core (as core_definiton) ----- *)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core reference usage') AND
  ('CAST_PARTS_SCHEMA.CORE' IN TYPEOF(sar.relating_shape_aspect))
  )) >=1 )))) >=1;

(* ---- production_core_box to core_equipment (as core_made_by)----- *)

wr9:(NOT(SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.ACTION_METHOD'
  IN TYPEOF(adr.assigned_action_method)) AND
  (SIZEOF(QUERY(ar <* USEDIN(adr.assigned_action_method,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('CAST_PARTS_SCHEMA.CORE_EQUIPMENT' IN TYPEOF(ar) ))>=1))))=0));

```

```

(* -production_core_box to core_print (as core_print_definition) ----- *)

wr10:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'core print reference usage') AND
  ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=0 )))) >=1;

(* ----- production_core_box to drafted_surface (as draft) *)

wr11: SIZEOF (QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) )) ) >=1;

(* -production_core_box to loose_piece(as loose_piece_requirement) --- *)

wr12:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece reference usage') AND
  ('CAST_PARTS_SCHEMA.LOOSE_PIECE' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=0 )))) >=1;

(* production_core_box to in_facility_location (as core_box_location) *)

wr13:((SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) )) ) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
  (ada.role.name='core box stored')))=1)
  );

(* ----- production_core_box to item_size (as guide_pin) ----- *)

WR14:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'guide pin size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) <=1 )))) >=1;

(* - production_core_box to item_size (as outside_shape_of_core_box) --- *)

```

```

WR15:((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
AND
  (sar.description = 'outside size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1)
OR
(SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'outside size usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1));

(* - production_core_box to parting_surface (as cope_and_drag_parting) *)

wr16:SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='cope and drag parting')))) >=1;

(* - production_core_box to parting_surface (as offset_parting) --- *)

wr17: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='offset parting')))) >=1;

(* --- production_core_box to shape_element (as invest_area_shape)---- *)

WR18: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment area shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
  )) >=1 )))) >=1;

(* --- production_core_box to shape_element (as impression_shape)----- *)

WR19: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production core box occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'impression shape usage') AND

```

```

('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
)) >=1 ))) >=1;

(* * to machining_allowance (as machine_stock_on_tool) ** *)

wr20: (SIZEOF(QUERY (co <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co)))>=1);

END_ENTITY; -- production_core_box
(*)
production_die_cast_mould
*)
ENTITY production_die_cast_mould
SUBTYPE OF (production_tool);
WHERE

(* ***** port_connection_type ***** *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'port connection type')))) =1 ))))=1;

(* ***** insert_material_type ***** *)

wr2: (SIZEOF(QUERY (co <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (co)))<=1);

(* ----- cooling_requirement ----- *)

wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS')
IN TYPEOF(pdr.used_representation)) ) |
(NOT (SIZEOF(QUERY ( it <* impl_rep.used_representation.items |
(('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND (it.name = 'cooling requirement')
AND (it.description IN ['true','false'] )
)))=1))))=0))))=0);

(* ----- die_type ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'

```

```

    IN TYPEOF(it)) AND ((it.name = 'die type') AND
    ((it.description IN ['hot chamber', 'cold chamber', 'vacum chamber']
    )) )) =1 )) )>=1;

(* ----- inserts ----- *)

wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'insert') AND
    ((it.description IN ['unit', 'combination', 'cast in', 'none']
    )) )) ) ) <=1 )) )>=1;

(* production_die_cast_mould to item_size (as external_box_dimensions) *)

WR6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='production die cast mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
    AND
    (sar.description = 'production die cast mould dimension usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) =1 )))) >=1;

(* ***** production_die_cast_mould to sprue (as pouring_sprue) ***** *)

wr7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='production die cast mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'sprue reference usage') AND
    ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relying_shape_aspect))
    )) =1 )))) >=1;

(* ***** production_die_cast_mould to runner (as gate_runner) ***** *)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='production die cast mould occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'gate runner reference usage') AND
    ( SIZEOF(['CAST_PARTS_SCHEMA.RUNNER',
    'CAST_PARTS_SCHEMA.INGATE']
    * TYPEOF(sar.relying_shape_aspect)) =1))) >=1 ))))>=1;

(* production_die_cast_mould to slide (as movable_die_items) ***** *)

```

ISO 10303-223:2008 (E)

```
(* ** production_die_cast_mould to die_clamping (as movable_die_items)
*)

wr9: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='production die cast mould occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'movable die clamping reference usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.SLIDE',
'CAST_PARTS_SCHEMA.DIE_CLAMPING']
* TYPEOF(sar.relatinq_shape_aspect)) =1))) >=0 )))>=1;

(* *** production_die_cast_mould to cooling_port (as cooling_type) *****
*)

wr10: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='production die cast mould occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'cooling port reference usage') AND
('CAST_PARTS_SCHEMA.COOLING_PORT' IN TYPEOF(sar.relatinq_shape_aspect))
)) <=1 ))) >=1;

(* to secondary_tooling (as secondary_tooling_requirements) ***** *)

wr11:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='production die cast mould occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'secondary tooling reference usage') AND
('CAST_PARTS_SCHEMA.SECONDARY_TOOLING'
IN TYPEOF(sar.relatinq_shape_aspect))
)) >=0 ))) >=1;

(* production_die_cast_mould to die_clamping (as stationary_die_items) *)
(* production_die_cast_mould to shot_sleeve (as stationary_die_items) *)

wr12: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
((sa.description='production die cast mould occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'stationary die clamping reference usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.SHOT_SLEEVE',
'CAST_PARTS_SCHEMA.DIE_CLAMPING']
* TYPEOF(sar.relatinq_shape_aspect)) =1))) >=1 )))>=1;

END_ENTITY; -- production_die_cast_mould
(*
```

**production\_die\_mould**

1640

© ISO 2008—All rights reserved



```

*)
ENTITY production_die_mould
  SUBTYPE OF (production_tool);
WHERE

(* ***** die_type ***** *)

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'die type')))) =1 )))=1;

(* --- production_die_mould to item_size (as external_box_dimensions) -*)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar))
AND
  (sar.description = 'production die mould dimension usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relating_shape_aspect))
  )) =1 )))) >=1;

(* ***** production_die_mould to slide (as slide_requirement) ***** *)

wr3:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND
  (sar.description = 'slide reference usage') AND
  ('CAST_PARTS_SCHEMA.SLIDE' IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

(* --- production_die_mould to gating_system (as gate_rigging) -----
*)

wr4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(sar)) AND
  (sar.description = 'gating system reference usage') AND
  ('CAST_PARTS_SCHEMA.GATING_SYSTEM'
  IN
  TYPEOF(sar.relating_shape_aspect))
  )) =1 )))) >=1;

```

## ISO 10303-223:2008 (E)

```
(* ----- production_die_mould to die_core (as core for die) -----
*)
(* ---- production_die_mould to core(as core for die) ----- *)
```

```
wr5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production die mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'core reference usage') AND
    ( SIZEOF(['CAST_PARTS_SCHEMA.CORE',
      'CAST_PARTS_SCHEMA.DIE_CORE']
      * TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 )))>=1;
```

```
END_ENTITY; -- production_die_mould
```

```
(*
```

### **production\_investment\_cast\_mould**

```
*)
```

```
ENTITY production_investment_cast_mould
  SUBTYPE OF (production_tool);
  WHERE
```

```
(* * production_investment_cast_mould to slide (as slide_requirement) * *)
```

```
wr1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production investment cast mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'slide reference usage') AND
    ('CAST_PARTS_SCHEMA.SLIDE' IN TYPEOF(sar.relatng_shape_aspect))
    )) >=1 ))) >=1;
```

```
(* ----- to in_facility_location (as die_mould_location) *)
```

```
wr2: ((SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
  IN TYPEOF ( pdr.used_representation ) ) ) ) =1) OR
  (SIZEOF(QUERY(ada <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS') |
    (ada.role.name='core box stored'))=1)
  );
```

```
END_ENTITY; -- production_investment_cast_mould
```

```
(*
```

### **production\_tool**

```
*)
```

```
ENTITY production_tool
  SUBTYPE OF (die_design_feature);
  WHERE
```

```
(* -----shrink_factor----- *)
```

1642

© ISO 2008—All rights reserved

```

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it)) = 2) AND
  (it.name = 'shrink factor')) ) ) =1 )))=1;

(* ----- male_or_female_mould----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'male or female') AND
  ((it.description IN ['male mould', 'female mould'] ) ) ) ) =1 ) ) )>=1;

(* ----- number_of_cavities -----*)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of cavities')) ) ) =1 )))=1;

(* ----- production_tool to item_size (as guide_pin) ----- *)

WR4:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar))
AND
  (sar.description = 'guide pin shape usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relying_shape_aspect))
  ) ) <=1 ))) ) >=1;

(* ----- production_tool to die_core (as core_for_die) ----- *)

wr5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'die core reference usage') AND
  ('CAST_PARTS_SCHEMA.DIE_CORE' IN TYPEOF(sar.relying_shape_aspect))

```

ISO 10303-223:2008 (E)

```

    )) >=0 )))) >=1;

(* -- production_tool to ejector_system (as ejection_definition) ---- *)

wr6:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ejector system reference usage') AND
  ('CAST_PARTS_SCHEMA.EJECTOR_SYSTEM'
  IN TYPEOF(sar.relying_shape_aspect))
  )) =1 )))) >=1;

(* - production_tool to production_core_box (as core_definition) --- *)

wr7:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production tool occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production core box reference usage') AND
  ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
  IN TYPEOF(sar.relying_shape_aspect))
  )) >=0 )))) >=1;

(* * to machining_allowance (as allowance_on_machinable_surfaces) ** *)

wr8: (SIZEOF(QUERY (co <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
  'CAST_PARTS_SCHEMA.MACHINE_ALLOWANCE' IN TYPEOF (co))<=1);

(* - production_tool to parting_surface (as male_or_female_parting) - *)

wr9: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='male or female parting')))) >=1;

(* - production_tool to parting_surface (as offset_parting_surfaces) *)

wr10: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation )) AND
  (pdr.used_representation.name='offset parting')))) >=0;

(* production_tool to drafted_surface (as draft) *)

wr11: SIZEOF (QUERY(pdr <* get_property_definition_representations (SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) )) >=1;

(* ----- to shape_aspect (as outside_shape_of_die_mould)----- *)

```

```
(* ----- production_tool to (as outside_shape_of_die_mould) -----
*)
```

```
WR12: ((SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='production tool occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'production tool outside shape usage') AND
('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
IN TYPEOF(sar.relater_shape_aspect))
)) =1 ))) >=1)
OR
(SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='production tool occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'production tool outside shape usage') AND
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relater_shape_aspect))
)) =1 ))) >=1));
```

```
(* --- production_tool to shape_element (as mould_impression_shape)-----
*)
```

```
WR13:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='production tool occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'production tool impression shape usage') AND
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relater_shape_aspect))
)) >=1 ))) >=1;
```

```
(* ----- production_tool to riser (as riser_on_mould) ----- *)
(* ---- production_tool to riser_contact (as riser_on_mould) ----- *)
```

```
wr14: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='production tool occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'riser reference usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.RISER',
'CAST_PARTS_SCHEMA.RISER_CONTACT']
* TYPEOF(sar.relater_shape_aspect)) =1))) >=0 )))>=1;
```

```
(* ----- production_tool to vent ( as vent requirement) ----- *)
```

```
WR15: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='production tool occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'vent reference usage') AND
('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relatng_shape_aspect))
)) >=0 )))) >=1;

END_ENTITY; -- production_tool
(*)
production_pattern_definition
*)
ENTITY production_pattern_definition
  SUBTYPE OF (sand_cast_design_feature);
  WHERE
(*) ***** condition ***** *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'condition') AND
  (it.description IN ['good','bad','fair','new'] )) ) ) =1 )))) >=1;

(*) ----- number_of_patterns_mounted ----- *)

wr2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'number of patterns mounted')) ) ) =1 ))))=1;

(*) -----shrink_factor----- *)

wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
  * TYPEOF(it) ) = 2) AND
  (it.name = 'shrink factor')) ) ) =1 ))))=1;

(*) ---production_pattern_definition to casting_insert (as inserts) -- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'casting insert reference usage') AND

```

```

('CAST_PARTS_SCHEMA.CASTING_INSERT'
 IN TYPEOF(sar.relatng_shape_aspect))
)) >=0 )))) >=1;

(* -- to chaplet_pad (as chaplet_pad_definition) -- *)

wr5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
 ((sa.description='production pattern definition occurrence') AND
 (SIZEOF(QUERY (sar <* USEDIN(sa,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
 (sar.description = 'chaplet pad reference usage') AND
 ('CAST_PARTS_SCHEMA.CHAPLET_PAD' IN TYPEOF(sar.relatng_shape_aspect))
 )) >=0 )))) >=1;

(* --production_pattern_definition to chill (as chill_pattern) --- *)

wr6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
 ((sa.description='production pattern definition occurrence') AND
 (SIZEOF(QUERY (sar <* USEDIN(sa,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
 (sar.description = 'chill reference usage') AND
 ('CAST_PARTS_SCHEMA.CHILL' IN TYPEOF(sar.relatng_shape_aspect))
 )) >=0 )))) >=1;

(* -- to in_facility_location (as pattern_location) *)

wr7: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELf) |
 ( 'CAST_PARTS_SCHEMA.IN_FACILITY_LOCATION'
 IN TYPEOF ( pdr.used_representation ) )) =1;

(* -- to production_core_box(as cores_for_pattern) --*)

wr8: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
 ((sa.description='production pattern definition occurrence') AND
 (SIZEOF(QUERY (sar <* USEDIN(sa,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
 (sar.description = 'production core box reference usage') AND
 ('CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX'
 IN TYPEOF(sar.relatng_shape_aspect))
 )) >=0 )))) >=1;

(* -- to core_print (as core_print_definition) -- *)

wr9: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
 ((sa.description='production pattern definition occurrence') AND
 (SIZEOF(QUERY (sar <* USEDIN(sa,
 'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
 (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
 (sar.description = 'core print reference usage') AND
 ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(sar.relatng_shape_aspect))
 )) >=0 )))) >=1;

```

ISO 10303-223:2008 (E)

```

    )) >=0 )))) >=1;

(* ---- production_pattern_definition to drafted_surface (as draft) *)

wr10: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  ( 'CAST_PARTS_SCHEMA.DRAFTED_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ))) >=1;

(* - to loose_piece(as loose_piece_requirement) -- *)

wr11: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'loose piece reference usage') AND
  ('CAST_PARTS_SCHEMA.LOOSE_PIECE' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=0 )))) >=1;

(* --to parting_surface (as cope_and_drag_parting) - *)

wr12: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ) AND
  (pdr.used_representation.name='cope and drag parting')))) =1;

(* -- to parting_surface (as offset_parting) ----- *)

wr13: SIZEOF(QUERY(pdr <* get_property_definition_representations(SELF) |
  (( 'CAST_PARTS_SCHEMA.PARTING_SURFACE'
  IN TYPEOF ( pdr.used_representation ) ) AND
  (pdr.used_representation.name='offset parting')))) >=0;

(* -- to pattern_plate (as match_plate_patterns) ---- *)

wr14: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'pattern plate reference usage') AND
  ('CAST_PARTS_SCHEMA.PATTERN_PLATE'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) >=0 )))) >=1;

(* ----- to riser (as riser_on_pattern) ----- *)
(* ----- to riser_contact (as riser_on_pattern) ----- *)

wr15: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND

```



```

(sar.description = 'riser reference usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.RISER',
         'CAST_PARTS_SCHEMA.RISER_CONTACT']
        * TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 ))))>=1;

(* ----- to shape_element (as invest_area)----- *)

WR16:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='production pattern definition occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'production pattern definition shape usage') AND
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=1 )))) >=1;

```

```

END_ENTITY; -- production_pattern_definition
(*

```

## **production\_rate**

```

*)
ENTITY production_rate
  SUBTYPE OF (action_property);
  WHERE
  wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF
        (it)) = 2) AND (it.name = 'time per unit')))) = 1))) = 1;

  wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'unit quantity'))))=1)))=1;

  wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (SIZEOF(QUERY ( it <* apr.representation.items |
      (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'production rate source'))))=1)))=1;

  END_ENTITY; -- production_rate
(*

```

## **production\_welding**

```

*)
ENTITY production_welding
  SUBTYPE OF (non_casting_activity);
  END_ENTITY; -- production_welding
(*

```

## **profile\_floor**

\*)

```

ENTITY profile_floor
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
      SELF.of_shape.definition);
    wr2 : SELF.description IN ['planar','complex','through'];
    wr3 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0);
    wr4 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) | ((NOT (SIZEOF(impl_rep.
        used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
        used_representation.items) <= 2)) )) = 0)) )) = 0);
    wr5 : (NOT (SELF.description = 'through')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 0)) )) = 0);
    wr6 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) )) = 0);
    wr7 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | ((sar.description =
      'profile floor usage') AND (
      'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) )) | (('CAST_PARTS_SCHEMA.OUTSIDE_PROFILE' IN

```

```

        TYPEOF(fcr.related_shape_aspect.of_shape.definition)) AND (
        'CAST_PARTS_SCHEMA.PROFILE_FLOOR' IN TYPEOF(fcr.
        relating_shape_aspect))) ) ) >= 1;
wr8 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
        QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name =
        'shape profile floor orientation')) AND (it.description IN
        ['shape profile start','shape profile end'])) ) = 1)) ) =
        0)) ) = 0);
wr9 : (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr
        .used_representation)) AND (pdr.used_representation.name =
        'floor')) ) = 1)) ) = 1);
wr10: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
        pdr.used_representation)) AND (pdr.used_representation.name
        = 'floor')) ) = 1)) ) = 1);
END_ENTITY; -- profile_floor

```

(\*

## projected\_zone\_definition

\*)

```

ENTITY projected_zone_definition
  SUBTYPE OF (tolerance_zone_definition);
  projection_end      : shape_aspect;
  projected_length    : measure_with_unit;
WHERE
  wr1: projected_length.value_component > 0;
END_ENTITY; -- projected_zone_definition

```

(\*

## property\_definition

\*)

```

ENTITY property_definition;
  name          : label;
  description    : OPTIONAL text;
  definition     : characterized_definition;
DERIVE

```

```

    id : identifier := get_id_value(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- property_definition

```

(\*

### **property\_definition\_relationship**

\*)

```

ENTITY property_definition_relationship;
    name                : label;
    description          : text;
    relating_property_definition : property_definition;
    related_property_definition  : property_definition;
END_ENTITY; -- property_definition_relationship

```

(\*

### **property\_definition\_representation**

\*)

```

ENTITY property_definition_representation;
    definition          : represented_definition;
    used_representation : representation;
DERIVE
    description : text := get_description_value(SELF);
    name        : label := get_name_value(SELF);
WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- property_definition_representation

```

(\*

### **property\_inspection**

\*)

```

ENTITY property_inspection
    SUBTYPE OF (inspection_or_test_result);
WHERE
    (* --to property_parameter (as property_result)--- *)

```

```

wr1:(NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY( it <* pdr.used_representation.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM']
    * TYPEOF(it)) = 1) AND
    (it.name = 'property result')) =1 ))) =0));

```

(\* --to property\_parameter (as requirements)---- \*)

```

wr2: (NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY( it <* pdr.used_representation.items |
    (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM']

```

```

    * TYPEOF(it)) = 1) AND
    (it.name = 'requirements') )) >=1))) =0));

END_ENTITY; -- property_inspection
(*)
property_process
*)
ENTITY property_process
  SUBTYPE OF (action);
  identification : identifier;
  INVERSE
  properties : SET [1:?] OF process_property_association FOR process;
END_ENTITY; -- property_process
(*)
property_relationship
*)
ENTITY property_relationship
  SUBTYPE OF (representation);
  WHERE
  (* --- x property name----- *)

wr1: SIZEOF(QUERY ( mri <* SELF.items | ((
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  (mri.name = 'x property name')) )) = 1;

  (* --- y property name----- *)

wr2: SIZEOF(QUERY ( mri <* SELF.items | ((
  'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(mri)) AND
  (mri.name = 'y property name')) )) = 1;

  (* --- property_relationship to data_curve (as data_points)----- *)

wr3: SIZEOF(QUERY ( rep <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.REPRESENTATION_RELATIONSHIP.REP_1') |
  (('CAST_PARTS_SCHEMA.DATA_CURVE' IN TYPEOF(rep.rep_2)) AND
  (rep.rep_2.name='relationship curve')))) >= 1;

END_ENTITY; -- property_relationship
(*)
protrusion
*)
ENTITY protrusion
  SUBTYPE OF (feature_definition);
  WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1)) ) = 0)) ) = 0;
wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'shape volume occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'volume shape usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (sdr.relatng_shape_aspect.description =
'volume shape') )) = 1)) ) = 1)) ) = 0;
wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit')) ) ) >= 0;
END_ENTITY; -- protrusion

```

(\*

### qualified\_representation\_item

\*)

```

ENTITY qualified_representation_item
  SUBTYPE OF (representation_item);
  qualifiers : SET [1:?] OF value_qualifier;
  WHERE
    wr1: SIZEOF(QUERY ( temp <* qualifiers | (
      'CAST_PARTS_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp)) ) )
      < 2;
END_ENTITY; -- qualified_representation_item

```

(\*

### quasi\_uniform\_curve

\*)

```

ENTITY quasi_uniform_curve
  SUBTYPE OF (b_spline_curve);
END_ENTITY; -- quasi_uniform_curve

```

(\*

### quasi\_uniform\_surface

\*)

```

ENTITY quasi_uniform_surface
  SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

```

(\*

### ratio\_measure\_with\_unit

\*)

```

ENTITY ratio_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.RATIO_UNIT' IN TYPEOF(SELF\measure_with_unit
      .unit_component);
END_ENTITY; -- ratio_measure_with_unit

```

(\*)

**ratio\_unit**

\*)

```

ENTITY ratio_unit
  SUBTYPE OF (named_unit);
  WHERE
    wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\named_unit.dimensions.mass_exponent = 0)) AND (SELF\named_unit.dimensions.time_exponent = 0)) AND (SELF\named_unit.dimensions.electric_current_exponent = 0)) AND (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent = 0)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- ratio_unit

```

(\*)

**rational\_b\_spline\_curve**

\*)

```

ENTITY rational_b_spline_curve
  SUBTYPE OF (b_spline_curve);
  weights_data : LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:upper_index_on_control_points] OF REAL :=
      list_to_array(weights_data,0,
        upper_index_on_control_points);
  WHERE
    wr1: SIZEOF(weights_data) = SIZEOF(SELF\b_spline_curve.control_points_list);
    wr2: curve_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_curve

```

(\*)

**rational\_b\_spline\_surface**

\*)

```

ENTITY rational_b_spline_surface
  SUBTYPE OF (b_spline_surface);
  weights_data : LIST [2:?] OF LIST [2:?] OF REAL;
  DERIVE
    weights : ARRAY [0:u_upper] OF ARRAY [0:v_upper] OF REAL :=
      make_array_of_array(weights_data,0,u_upper,0,v_upper);
  WHERE
    wr1: (SIZEOF(weights_data) = SIZEOF(SELF\b_spline_surface.control_points_list)) AND (SIZEOF(weights_data[1]) = SIZEOF(SELF\b_spline_surface.control_points_list[1]));
    wr2: surface_weights_positive(SELF);
END_ENTITY; -- rational_b_spline_surface

```

(\*)

**rectangular\_closed\_profile**

\*)

```

ENTITY rectangular_closed_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF

```

```

        .of_shape.definition);
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT ((SIZEOF(impl_rep.
    used_representation.items) >= 3) AND (SIZEOF(impl_rep.
    used_representation.items) <= 4))) )) = 0)) )) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
    USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND (SIZEOF(
    QUERY ( srwp_i <* pdr.used_representation.items | (((srwp_i
    .name = 'orientation') OR (srwp_i.name = 'length')) OR (
    srwp_i.name = 'width')) OR (srwp_i.name = 'corner radius')) ))
    = SIZEOF(pdr.used_representation.items))) )) = 1)) )) = 1;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((
    'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
    'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (

```



```

SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'length')) ) = 1)) ) = 0)) ) = 0;
wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'corner radius')) ) <= 1)) ) = 0)) ) )
= 0;
END_ENTITY; -- rectangular_closed_profile
(*)
rectangular_pattern
*)
ENTITY rectangular_pattern
SUBTYPE OF (replicate_feature);
WHERE
wr1 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATING_SHAPE_ASPECT') | (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar)) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN
TYPEOF(sdr.related_shape_aspect)) ) = 1)) <= 5)) ) = 0;
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'row layout direction')) ) = 1)) ) )
= 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

        'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation) AND (pdr.
        used_representation.name = 'column layout direction')) )) =
        1)) )) = 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.'
        + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 5)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
        it) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
        measure_with_unit.value_component))) AND (it.name =
        'number of rows')) )) = 1)) )) = 0)) )) = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
        it) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
        measure_with_unit.value_component))) AND (it.name =
        'number of columns')) )) = 1)) )) = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'row spacing')) ) = 1)) ) = 0)) ) =
0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'column spacing')) ) = 1)) ) = 0)) ) )
= 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) ) = 1)) ) = 0)) ) = 0;
END_ENTITY; -- rectangular_pattern
(*
red_lined_part
*)
ENTITY red_lined_part
SUBTYPE OF (product_definition);
WHERE

(* ----- red_lined_part to feature (as mark_up_feature) *)

wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='red lined part occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'red lined feature usage') AND
( SIZEOF(['CAST_PARTS_SCHEMA.INSTANCED_FEATURE',
'CAST_PARTS_SCHEMA.CASTING_INSTANCED_FEATURE']
* TYPEOF(sar.relatng_shape_aspect)) =1))) >=0 )))>=1;

(* ----- red_lined_part to shape_aspect (as mark_up_shape) *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |

```

```
((sa.description='red lined part occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'red lined shape usage') AND
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
)) >=0 ))) >=1;
```

```
(* ----- red_lined_part to material (as mark_up_material) *)
```

```
WR3: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.'+
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
(('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
(pdr.name='mark up material') AND
(SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
))>=0))))=0));
```

```
(* ----- red_lined_part to property (as mark_up_property) *)
```

```
wr4: SIZEOF(QUERY(pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(pd.id='mark up property')) >=0;
```

```
END_ENTITY; -- red_lined_part
```

```
(*
```

### **referenced\_modified\_datum**

```
*)
```

```
ENTITY referenced_modified_datum
SUBTYPE OF (datum_reference);
modifier : limit_condition;
END_ENTITY; -- referenced_modified_datum
```

```
(*
```

### **relationship\_condition**

```
*)
```

```
ENTITY relationship_condition;
name : label;
applicable_relationships : SET [1:?] OF relationship_with_condition;
condition_description : text;
END_ENTITY; -- relationship_condition
```

```
(*
```

### **removal\_volume**

```
*)
```

```
ENTITY removal_volume
SUBTYPE OF (feature_definition);
WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
```

```

        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) ) = 0)) ) = 0;
wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'shape volume occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'volume shape usage') AND (('CAST_PARTS_SCHEMA.' +
        'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (sdr.
        relating_shape_aspect.description = 'volume shape') )) = 1)) )
        = 1)) ) = 0;
END_ENTITY; -- removal_volume
(*)
replicate_feature
*)
ENTITY replicate_feature
  SUPERTYPE OF (ONEOF (circular_pattern, rectangular_pattern,
    feature_pattern))
  SUBTYPE OF (feature_definition);
WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) = 1)) ) = 0;
  wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar)) ) | ((SIZEOF([
    'CAST_PARTS_SCHEMA.REPLICATE_FEATURE',
    'CAST_PARTS_SCHEMA.INSTANCED_FEATURE'] * TYPEOF(fcr.
    related_shape_aspect)) >= 1) AND (fcr.name = 'pattern basis')) )
    = 1;
  wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | (NOT (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar))) ) + SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | (NOT (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN

```

```

        TYPEOF(sar))) ))) = 0;
    END_ENTITY; -- replicate_feature
    (*
reporting_requirement
    *)
    ENTITY reporting_requirement
        SUBTYPE OF (customer_casting_requirement);
        WHERE
    (* --- reporting_requirement to date (as date_to_be_reported)----- *)

    wr1: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
        (ada.role.name='required delivery date')))=1);

    (* reporting_requirement to customer_casting_requirement (as based_on)- *)

    wr2:(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.'+
        'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
        (('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT'
        IN TYPEOF (pdr.related_property_definition)
        )) ) ) =1);

    END_ENTITY; -- reporting_requirement
    (*

```

## representation

```

    *)
    ENTITY representation;
        name : label;
        items : SET [1:?] OF representation_item;
        context_of_items : representation_context;
    DERIVE
        id : identifier := get_id_value(SELF);
        description : text := get_description_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
        wr2: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    END_ENTITY; -- representation
    (*

```

## representation\_context

```

    *)
    ENTITY representation_context;
        context_identifier : identifier;
        context_type : text;
    INVERSE
        representations_in_context : SET [1:?] OF representation FOR
            context_of_items;
    END_ENTITY; -- representation_context
    (*

```

## representation\_item

```

*)
ENTITY representation_item;
    name : label;
    WHERE
        wr1: SIZEOF(using_representations(SELF)) > 0;
END_ENTITY; -- representation_item
(*
representation_map
*)
ENTITY representation_map;
    mapping_origin      : representation_item;
    mapped_representation : representation;
    INVERSE
        map_usage : SET [1:?] OF mapped_item FOR mapping_source;
    WHERE
        wr1: item_in_context(SELF.mapping_origin, SELF.mapped_representation.
            context_of_items);
END_ENTITY; -- representation_map
(*
representation_relationship
*)
ENTITY representation_relationship;
    name      : label;
    description : OPTIONAL text;
    rep_1     : representation;
    rep_2     : representation;
END_ENTITY; -- representation_relationship
(*
request_for_quotation
*)
ENTITY request_for_quotation
    SUBTYPE OF (versioned_action_request);
    WHERE
        (* ----- quantity_breaks -----*)

wr1: SIZEOF(QUERY(ars <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
    (NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
    (NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
    (NOT (SIZEOF(QUERY(it <* apr.representation.items |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component))) AND
    (it.name = 'quantity_breaks'))
    ))>=1)) ))=0)) ))=0)) ))=0;

        (* ----- quantity_per_release -----*)

wr2: SIZEOF(QUERY(ars <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |

```

```
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity per release'))
)) <=1)) ))=0)) ))=0)) ))=0;
```

(\* ----- quantity\_per\_month -----\*)

```
wr3: SIZEOF(QUERY(ars <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity per month'))
)) <=1)) ))=0)) ))=0)) ))=0;
```

(\* ----- quantity\_per\_year -----\*)

```
wr4: SIZEOF(QUERY(ars <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
(it.name = 'quantity per year'))
)) <=1)) ))=0)) ))=0)) ))=0;
```

(\* ----- quantity\_per\_order -----\*)

```
wr5: SIZEOF(QUERY(ars <* USEDIN(SELf,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
((( 'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component))) AND
```



```

(it.name = 'total quantity per order'))
))=1)) ))=0)) ))=0)) ))=0;

(* ----- type_of_quotation -----*)

wr6: SIZEOF(QUERY(ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
(NOT (SIZEOF(QUERY(ap <* USEDIN(ars.method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(NOT (SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name = 'type of quotation'))
))=1)) ))=0)) ))=0)) ))=0;

(* - request_for_quotation to date (as delivery_date) - *)

wr7: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='delivery date')))=1);

(* -- request_for_quotation to date (as reply_due_date) - *)

wr8: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='reply due date')))=1);

(* -- request_for_quotation to date (as request_date) -*)

wr9: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='request date')))=1);

(* -- request_for_quotation to date (as order_completion) - *)

wr10: (SIZEOF(QUERY(ada <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
(ada.role.name='order completion')))=1);

END_ENTITY; -- request_for_quotation
(*)
requirement_for_action_resource
*)
ENTITY requirement_for_action_resource
SUBTYPE OF (action_resource_requirement);
resources : SET [1:?] OF action_resource;
END_ENTITY; -- requirement_for_action_resource
(*)
reset_or_push_pin
*)
ENTITY reset_or_push_pin

```

## ISO 10303-223:2008 (E)

```

SUBTYPE OF (ejector_design_feature);
WHERE

(* ***** spring_description ***** *)

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SEL) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name = 'spring_description')))) <=1 )))=1;

(* - reset_or_return_pins to item_size (as pin_dimensions) ----- *)

WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SEL) |
  ((sa.description='reset or push pin occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'reset or push pin dimensions size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relateing_shape_aspect))
  )) =1 ))) >=1;

END_ENTITY; -- reset_or_push_pin
(*
resource_property
*)
ENTITY resource_property;
  name          : label;
  description   : text;
  resource      : characterized_resource_definition;
END_ENTITY; -- resource_property
(*
resource_property_representation
*)
ENTITY resource_property_representation;
  name          : label;
  description   : text;
  property      : resource_property;
  representation : representation;
END_ENTITY; -- resource_property_representation
(*
resource_requirement_type
*)
ENTITY resource_requirement_type;
  name          : label;
  description   : text;
END_ENTITY; -- resource_requirement_type
(*
resource_with_material
*)
1666
```

© ISO 2008—All rights reserved

```

ENTITY resource_with_material
  SUBTYPE OF (generic_manufacturing_resource);
  WHERE
  wr1: (NOT (SIZEOF(QUERY( am <* SELF.usage |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
    (NOT(SIZEOF(QUERY ( ama <* USEDIN(am,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY ( eds <* ama.items |
    ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(eds))
    ))=1))))=0))))=0));

```

```

END_ENTITY; -- resource_with_material

```

(\*

## resource\_with\_representation

\*)

```

ENTITY resource_with_representation
  SUBTYPE OF (generic_manufacturing_resource);
  WHERE
  wr1:(NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (SIZEOF(QUERY ( rpr <* USEDIN(ap,
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
    ('CAST_PARTS_SCHEMA.EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF(rpr.representation)) )) = 1) )) = 0));

```

```

END_ENTITY; -- resource_with_representation

```

(\*

## revision

\*)

```

ENTITY revision
  SUBTYPE OF (action_relationship);
  WHERE
  (* ----- reason_for_revision ----- *)

```

```

wr1: SIZEOF(QUERY ( rc <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'RELATIONSHIP_CONDITION.APPLICABLE_RELATIONSHIPS') |
    (rc.name = 'reason for revision')) = 1;

```

```

(* ----- revision_level ----- *)

```

```

wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF.related_action.chosen_method,
    'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
    (('CAST_PARTS_SCHEMA.VERSIONED_ACTION_REQUEST'
    IN TYPEOF(ars.request)) AND
    (ars.request.description = 'revision level')))) = 1;

```

```

(* ----- revision to status_authority ( as approved_by) ----- *)

```

```

wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS') |

```

```

('CAST_PARTS_SCHEMA.APPROVAL'
 IN TYPEOF(adr.assigned_approval))) >= 1;

(* ----- revision to process_plan_version (as related_to ----- *)
(* ----- revision to process_plan_version (as relating_to ----- *)

wr4: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
 IN TYPEOF(SELF.related_action)) AND
('CAST_PARTS_SCHEMA.PROCESS_PLAN_VERSION'
 IN TYPEOF(SELF.relatng_action));

END_ENTITY; -- revision
(*)
revolved_profile
*)
ENTITY revolved_profile
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SELF\characterized_object.description IN ['groove','flat',
      'round','open profile'];
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 2)) ) = 0)) ) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
      = 2) AND (it.name = 'radius')) ) = 1)) ) = 0)) ) = 0;
    wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | (('CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION'
      IN TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'removal direction')) ) = 1;
    wr5: (NOT (SELF\characterized_object.description = 'open profile'))
      OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
      | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'outer edge shape occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,

```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE' IN
TYPEOF(sdr.relatng_shape_aspect)) AND (sdr.
relatng_shape_aspect.description = 'outer edge shape')) ))
= 1)) )) = 1)) )) = 0);
wr6: (NOT (SELF\characterized_object.description = 'flat')) OR (
SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'flat edge shape occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.LINEAR_PROFILE' IN TYPEOF(
sdr.relatng_shape_aspect)) AND (sdr.relatng_shape_aspect.
description = 'flat edge shape')) )) = 1)) )) = 1)) )) = 0);
wr7: (NOT (SELF\characterized_object.description = 'round')) OR (
SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'rounded edge shape occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
TYPEOF(sdr.relatng_shape_aspect)) AND (sdr.
relatng_shape_aspect.description = 'rounded edge shape')) ))
= 1)) )) = 1)) )) = 0);
wr8: (NOT (SELF\characterized_object.description = 'groove')) OR (
SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'sweep occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ((SIZEOF(['CAST_PARTS_SCHEMA.SQUARE_U_PROFILE',
'CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE',

```

```

        'CAST_PARTS_SCHEMA.ROUNDED_U_PROFILE',
        'CAST_PARTS_SCHEMA.VEE_PROFILE',
        'CAST_PARTS_SCHEMA.TEE_PROFILE',
        'CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
        relating_shape_aspect)) = 1) AND (sdr.relatng_shape_aspect.
        description = 'sweep')) ) = 1)) ) = 1)) ) = 0);
    wr9: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'maximum feature limit')) ) ) >= 0;
END_ENTITY; -- revolved_profile
(*
rib
*)
ENTITY rib
  SUBTYPE OF (casting_design_feature);
  WHERE

    wr1 : SELF\characterized_object.description IN ['planar','complex'];

(* ----- general_rib to numeric_parameter (as part_rib_fillet)----- *)

    wr2: ((NOT (SELF\characterized_object.description = 'complex')) OR
    (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'part rib fillet')) ) =1 )))) =1 )));

(* ----- general_rib to numeric_parameter (as wall_to_floor_fillet)- *)

    wr3: ((NOT (SELF\characterized_object.description = 'complex')) OR
    (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
    * TYPEOF(it)) = 2) AND
    (it.name = 'wall to floor fillet')) ) =1 )))) =1 )));

(* ----- general_rib to numeric_parameter (as wall_to_top_fillet)----- *)

    wr4: ((NOT (SELF\characterized_object.description = 'complex')) OR
    (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']

```

```

        * TYPEOF(it)) = 2) AND
        (it.name = 'wall to top radius')) )) =1 )))) =1 ));

(* ---- general_rib to face_shape_element(as walls_top_floor)----- *)
(* --- general_rib to planar_element(as walls_top_floor) ----- *)

WR5: ((NOT (SELF\characterized_object.description = 'complex')) OR
      (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
                    ((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
                              'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION']
                              * TYPEOF(pdr.used_representation)) = 1)AND
                    (pdr.used_representation.name='walls top floor'))))>=1));

(* -----planar_rib to numeric_parameter (as part_rib_radius)--- *)

wr6: ((NOT (SELF\characterized_object.description = 'planar')) OR
      (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
                    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                     TYPEOF(pdr.used_representation)) AND
                     (SIZEOF(QUERY (it <* pdr.used_representation.items |
                                     ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                                               'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
                                               * TYPEOF(it)) = 2) AND
                                     (it.name = 'part rib radius')) )) =1 )))) =1 ));

(* --Planar_rib to numeric_parameter (as rib_extent_fillet)- *)

wr7: ((NOT (SELF\characterized_object.description = 'planar')) OR
      (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
                    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                     TYPEOF(pdr.used_representation)) AND
                     (SIZEOF(QUERY (it <* pdr.used_representation.items |
                                     ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                                               'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
                                               * TYPEOF(it)) = 2) AND
                                     (it.name = 'rib extent fillet')) )) =1 )))) =1 ));

(* - Planar_rib to numeric_parameter (as transition_to_top_radius)----- *)

wr8: ((NOT (SELF\characterized_object.description = 'planar')) OR
      (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
                    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
                     TYPEOF(pdr.used_representation)) AND
                     (SIZEOF(QUERY (it <* pdr.used_representation.items |
                                     ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
                                               'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
                                               * TYPEOF(it)) = 2) AND
                                     (it.name = 'transition to top radius')) )) =1 )))) =1 ));

(* - Planar_rib to numeric_parameter (as wall_to_top_radius) *)

wr9: ((NOT (SELF\characterized_object.description = 'planar')) OR
      (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |

```

```

        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'wall to top radius')) ) = 1 ))) = 1 ));

(* - Planar_rib to numeric_parameter (as wall_to_floor_fillet)----- *)

wr10: ((NOT (SELF\characterized_object.description = 'planar')) OR
        (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        ((SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT']
        * TYPEOF(it)) = 2) AND
        (it.name = 'wall to floor fillet')) ) = 1 ))) = 1 ));

(* ----- Planar_rib to face_shape_element(as top)----- *)
(* ----- Planar_rib to planar_element(as top) ----- *)
wr11: ((NOT (SELF\characterized_object.description = 'planar')) OR
        (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        ((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
        'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION']
        * TYPEOF(pdr.used_representation)) = 1)AND
        (pdr.used_representation.name='top'))))>=1));

(* ----- Planar_rib to planar_element(as walls) ----- *)

WR12: ((NOT (SELF\characterized_object.description = 'planar')) OR
        (SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        ((SIZEOF(['CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION']
        * TYPEOF(pdr.used_representation)) = 1)AND
        (pdr.used_representation.name='walls'))))>=1));
END_ENTITY; -- rib

```

**rib\_top**

```

*)
ENTITY rib_top
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'rib top condition occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'rib top usage') AND (

```



```

        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.RIB_TOP_FLOOR' IN
        TYPEOF(sdr.relatng_shape_aspect)) AND (
        'CAST_PARTS_SCHEMA.RIB_TOP' IN TYPEOF(sdr.
        related_shape_aspect.of_shape.definition))) ) = 1)) ) = 1)) )
        = 0;
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'removal direction')) ) = 1)) )
        = 1;
wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'maximum feature limit')) ) ) >= 0;
END_ENTITY; -- rib_top
(*)
rib_top_floor
*)
ENTITY rib_top_floor
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
        .of_shape.definition);
    wr2: SELF.description IN ['planar', 'complex'];
    wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') | ((sar.description =
        'rib top usage') AND (
        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
        'rib top condition occurrence') AND (
        'CAST_PARTS_SCHEMA.RIB_TOP' IN TYPEOF(fcr.
        related_shape_aspect.of_shape.definition))) AND (
        'CAST_PARTS_SCHEMA.RIB_TOP_FLOOR' IN TYPEOF(fcr.
        relating_shape_aspect))) ) ) >= 1;
    wr4: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION' IN TYPEOF(pdr.
        used_representation)) AND (pdr.used_representation.name =
        'rib top face')) ) = 1)) ) = 0);
    wr5: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
        USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(

```

```

        pdr.used_representation)) AND (pdr.used_representation.name
        = 'rib top face')) )) = 1)) )) = 0);
wr6: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pds <*
        QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'boundary occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | ((SIZEOF([
        'CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE',
        'CAST_PARTS_SCHEMA.NGON_CLOSED_PROFILE',
        'CAST_PARTS_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
        'CAST_PARTS_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
        relating_shape_aspect)) = 1) AND (sdr.relatng_shape_aspect.
        description = 'rib top floor boundary')) )) = 1)) )) = 1)) ))
        = 0);
END_ENTITY; -- rib_top_floor

```

(\*

**riser**

\*)

```

ENTITY riser
  SUBTYPE OF (gating_design_feature);
  WHERE

```

(\* \*\*\*\*\* riser\_type \*\*\*\*\* \*)

```

WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'riser type') AND
  ((it.description IN ['open','blind','open with radiation shield',
  'top','top_with_radiation_sheld'] )))))=1 ))))>=1;

```

(\* \*\*\*\*\* riser\_sleeve \*\*\*\*\* \*)

```

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND ((it.name = 'riser sleeve') AND
  ((it.description IN ['insulating','exothermic',
  'insulating and exothermic'] )))))<=1 ))))>=1;

```

(\* ----- risers to item\_size (as riser\_dimensions) ----- \*)

```

WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relating_shape_aspect))
  )) =1 )))) >=1;

```

```
(* ----- risers to item_size (as riser_neck) ----- *)
```

```

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser neck size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- risers to riser_contact (as contact_definition) ----- *)
```

```

WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'riser contact reference usage') AND
  ('CAST_PARTS_SCHEMA.RISER_CONTACT'
  IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- risers to well (as riser_well) ----- *)
```

```

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'well reference usage') AND
  ('CAST_PARTS_SCHEMA.WELL' IN TYPEOF(sar.relating_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- risers to item_size (as riser_core) ----- *)
```

```

WR7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='riser occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND

```

```

        (sar.description = 'riser core size usage') AND
        ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
         IN TYPEOF(sar.relatng_shape_aspect))
        )) <=1 )))) >=1;
    END_ENTITY; -- riser
(*)
riser_contact
*)
    ENTITY riser_contact
        SUBTYPE OF (gating_design_feature);
        WHERE

        (* ----- contact_placement ----- *)

        WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
            (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
             TYPEOF(pdr.used_representation)) AND
            (SIZEOF(QUERY (it <* pdr.used_representation.items |
                (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
                 IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
                ((it.description IN ['cope', 'drag', 'cope and drag'] )
                )) ) =1 ) ) ) >=1;

        (* ----- riser_contact to shape_element (as break_off_connection)----- *)

        wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
            ((sa.description='riser contact occurrence') AND
            (SIZEOF(QUERY (sar <* USEDIN(sa,
                'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
            (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
            (sar.description = 'riser contact shape usage') AND
            ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relatng_shape_aspect))
            )) <=0 )))) <=1;

        END_ENTITY; -- riser_contact
(*)
role_association
*)
    ENTITY role_association;
        role          : object_role;
        item_with_role : role_select;
    END_ENTITY; -- role_association
(*)
round_hole
*)
    ENTITY round_hole
        SUBTYPE OF (feature_definition);
        WHERE
            wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
                'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
                | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,

```

```

'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'diameter occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
'diameter'))) ) = 1)) ) = 1)) ) = 0;
wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'hole depth occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
'hole depth')) AND (sdr.relating_shape_aspect.description =
'linear'))) ) = 1)) ) = 1)) ) = 0;
wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'bottom condition occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'hole bottom usage') AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.HOLE_BOTTOM' IN
TYPEOF(fcr.relating_shape_aspect)) AND (
'CAST_PARTS_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
related_shape_aspect.of_shape.definition))) ) = 1)) ) = 1)) )
= 0;
wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'change in diameter occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
AND ('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.TAPER' IN TYPEOF(fcr.
relating_shape_aspect)) AND ('CAST_PARTS_SCHEMA.ROUND_HOLE'

```

```

        IN TYPEOF(fcr.related_shape_aspect.of_shape.definition))) ))
        = 1)) )) <= 1)) )) = 0;
wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) )) = 1;
END_ENTITY; -- round_hole
(*)
rounded_end
*)
ENTITY rounded_end
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'partial circular boundary occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | ('CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
        TYPEOF(sdr.relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'course of travel occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'path feature component usage') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | (('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
        TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
        relating_shape_aspect.description = 'linear')) )) = 1)) )) =
        1)) )) = 0;
    wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.

```

```

        used_representation.name = 'maximum feature limit')) )) >= 0;
END_ENTITY; -- rounded_end
(*)
rounded_u_profile
*)
ENTITY rounded_u_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
      .of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | ((NOT (SIZEOF(impl_rep.
      used_representation.items) >= 1)) AND (SIZEOF(impl_rep.
      used_representation.items) <= 2)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
      used_representation.items | ((srwp_i.name = 'orientation')
      OR (srwp_i.name = 'depth')) )) = SIZEOF(pdr.
      used_representation.items))) )) = 1)) )) = 1;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
      = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
      SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
        pdr.used_representation)) AND (pdr.used_representation.name
        = 'profile limit')) ) ) <= 1)) ) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'depth')) ) ) <= 1)) ) = 0)) ) = 0;
END_ENTITY; -- rounded_u_profile

```

(\*

### **roundness\_tolerance**

\*)

```

ENTITY roundness_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: NOT (('CAST_PARTS_SCHEMA.' +
              'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- roundness_tolerance

```

(\*

### **runner**

\*)

```

ENTITY runner
  SUBTYPE OF (gating_design_feature);
  WHERE

  (* ----- runner_placement ----- *)

  WR1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'cope or drag') AND
    ((it.description IN ['cope', 'drag', 'cope and drag'] )
    )) ) ) =1 ) ) )>=1;

```

(\* ----- runners to feeders (as riser\_connection) ----- \*)

```

  WR2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='runner occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'feeder reference usage') AND
    ('CAST_PARTS_SCHEMA.FEEDER' IN TYPEOF(sar.relater_shape_aspect))
    )) >=0 )))) >=1;

```



```
(* ----- runners to item_size (as runner_dimensions) ----- *)

WR3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'runner dimensions size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) =1 ))) >=1;

(* ----- runners to sprue ( as sprue_connections) ----- *)

WR4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue reference usage') AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=1 ))) >=1;

(* ----- runners to filter ( as metal_flow_filter) ----- *)

WR5: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'filter reference usage') AND
  ('CAST_PARTS_SCHEMA.FILTER' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* ----- runners to choke ( as runner choke) ----- *)

WR6: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'choke reference usage') AND
  ('CAST_PARTS_SCHEMA.CHOKE' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 ))) >=1;

(* ----- runners to vent ( as runner vent) ----- *)

WR7: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='runner occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'vent reference usage') AND
```

```

        ('CAST_PARTS_SCHEMA.VENT' IN TYPEOF(sar.relatng_shape_aspect))
        )) >=0 ))) >=1;

    END_ENTITY; -- runner
(*)
runout_zone_definition
*)
    ENTITY runout_zone_definition
        SUBTYPE OF (tolerance_zone_definition);
        orientation : runout_zone_orientation;
    END_ENTITY; -- runout_zone_definition
(*)
runout_zone_orientation
*)
    ENTITY runout_zone_orientation;
        angle : measure_with_unit;
    END_ENTITY; -- runout_zone_orientation
(*)
runout_zone_orientation_reference_direction
*)
    ENTITY runout_zone_orientation_reference_direction
        SUBTYPE OF (runout_zone_orientation);
        orientation_defining_relationship : shape_aspect_relationship;
    END_ENTITY; -- runout_zone_orientation_reference_direction
(*)
sampled_set
*)
    ENTITY sampled_set
        SUBTYPE OF (property_definition);
        WHERE
    (* ----- size----- *)

    wr1:(NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
        (it.name = 'size') )) <=1)) ) =0));

    (* ----- sampled_set to master_sample(as master_sample_inspected)-- *)

    (* **** SEE MASTER_SAMPLE FOR THIS RULE **** *)

    (* ----- sampled_set to heat(as sampled_form)----- *)
    (* ----- sampled_set to lot(as sampled_form)----- *)

    Wr2: (NOT(SIZEOF(QUERY ( da <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.APPLIED_EFFECTIVITY_ASSIGNMENT.ITEMS') |
        (SIZEOF([
        'CAST_PARTS_SCHEMA.LOT_EFFECTIVITY',
        'CAST_PARTS_SCHEMA.DATED_EFFECTIVITY'] *
        TYPEOF(da\effectivity_assignment.assigned_effectivity)) = 1)))=0));

```

```

(* -- to product_quality_report (as quality_assurance_document)- *)

Wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  (adr.assigned_document.kind.product_data_type=
  'production quality report'))
  >= 1;

  END_ENTITY; -- sampled_set
(*
sand_cast_design_feature
*)
ENTITY sand_cast_design_feature
  SUBTYPE OF (casting_feature_definition);
  WHERE

  WR1: SIZEOF(QUERY( pdr <*
    get_property_definition_representations (SELF) |
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
    IN TYPEOF ( pdr.used_representation ) ) ) =1;

(* ***** Sand_cast_design_feature to orientation (as placement) ***** *)

WR2: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
  (it.name = 'orientation')))) =1 )))=1;

  END_ENTITY; -- sand_cast_design_feature
(*
sand_casting_tooling
*)
ENTITY sand_casting_tooling
  SUBTYPE OF (casting_product_definition);
  WHERE

(* ----- sand_casting_tooling to material (as material_definition) *)

WR1: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
  (('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
  (pdr.name='material_definition') AND
  (SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
  'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF (mfuo)
  ))=1))))=0));

(* ----- sand_casting_tooling to material (as tooling_material) *)

```

```

WR2: (NOT(SIZEOF(QUERY (pdr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.'+
'PRODUCT_DEFINITION_RELATIONSHIP.RELATING_PRODUCT_DEFINITION') |
(('CAST_PARTS_SCHEMA.MAKE_FROM_USAGE_OPTION' IN TYPEOF(pdr)) AND
(pdr.name='tooling material') AND
(SIZEOF(QUERY(mfuo <* USEDIN(pdr.related_product_definition,
'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
(mfuo.name IN ['wood','cast iron','urethane','aluminum'] )
))=1))))=0));

(* to production_pattern_definition (as pattern_equipment) *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='sand casting tooling occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'production pattern definition reference usage') AND
('CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION'
IN TYPEOF(sar.relating_shape_aspect))
)) >=1 )))) >=1;

END_ENTITY; -- sand_casting_tooling
(*)
sand_mould
*)
ENTITY sand_mould
SUBTYPE OF (sand_cast_design_feature);
WHERE

(* ----- sand_mould_type ----- *)

WR1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
(('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation)) AND
(SIZEOF(QUERY (it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND ((it.name = 'sand mould type') AND
((it.description IN ['grean sand material','skin dried method',
'cold box mould method','no bake mould method'] )) )) ) =1 )) ))>=1;

(* ***** sand_mould to chaplet (as chaplet_definition) ***** *)

wr2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
((sa.description='sand mould occurrence') AND
(SIZEOF(QUERY (sar <* USEDIN(sa,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
(('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
(sar.description = 'chaplet reference usage') AND
('CAST_PARTS_SCHEMA.CHAPLET' IN TYPEOF(sar.relating_shape_aspect))
)) >=0 )))) >=1;

(* ***** sand_mould to chill (as chill_definition) ***** *)

```

```

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sand mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'chill reference usage') AND
  ('CAST_PARTS_SCHEMA.CHILL' IN TYPEOF(sar.relatng_shape_aspect))
  )) >=0 )))) >=1;

```

```
(* ----- sand_mould to equipment (as mould_made_by) *)
```

```

wr4: (NOT (SIZEOF(QUERY ( adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT.ITEMS') |
  (('CAST_PARTS_SCHEMA.ACTION_METHOD'
  IN TYPEOF(adr.assigned_action_method)) AND
  (SIZEOF(QUERY(ar <* USEDIN(adr.assigned_action_method,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('CAST_PARTS_SCHEMA.CASTING_EQUIPMENT' IN TYPEOF(ar) )))=1))))=0));

```

```
END_ENTITY; -- sand_mould
```

```
(*
```

## secondary\_tooling

```
*)
```

```

ENTITY secondary_tooling
  SUBTYPE OF (die_design_feature);
  WHERE

```

```
(* ***** type_of_tooling ***** *)
```

```

wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'type of tooling')))) =1 ))))=1;

```

```
END_ENTITY; -- secondary_tooling
```

```
(*
```

## security\_classification

```
*)
```

```

ENTITY security_classification;
  name : label;
  purpose : text;
  security_level : security_classification_level;
END_ENTITY; -- security_classification

```

```
(*
```

## security\_classification\_assignment

```
*)
```

```

ENTITY security_classification_assignment
  ABSTRACT SUPERTYPE;
  assigned_security_classification : security_classification;

```

```

    DERIVE
      role : object_role := get_role(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
    END_ENTITY; -- security_classification_assignment
  (*)
security_classification_level
  *)
    ENTITY security_classification_level;
      name : label;
    END_ENTITY; -- security_classification_level
  (*)
sequential_method
  *)
    ENTITY sequential_method
      SUBTYPE OF (serial_action_method);
      sequence_position : count_measure;
    END_ENTITY; -- sequential_method
  (*)
serial_action_method
  *)
    ENTITY serial_action_method
      SUBTYPE OF (action_method_relationship);
    END_ENTITY; -- serial_action_method
  (*)
shakeout
  *)
    ENTITY shakeout
      SUBTYPE OF (casting_equipment);
    END_ENTITY; -- shakeout
  (*)
shape_aspect
  *)
    ENTITY shape_aspect;
      name : label;
      description : OPTIONAL text;
      of_shape : product_definition_shape;
      product_definitional : LOGICAL;
    DERIVE
      id : identifier := get_id_value(SELF);
    WHERE
      wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
    END_ENTITY; -- shape_aspect
  (*)
shape_aspect_deriving_relationship
  *)
    ENTITY shape_aspect_deriving_relationship
      SUBTYPE OF (shape_aspect_relationship);
    WHERE

```

```

        wr1: 'CAST_PARTS_SCHEMA.DERIVED_SHAPE_ASPECT' IN TYPEOF(SELF\
            shape_aspect_relationship.relate_shape_aspect);
    END_ENTITY; -- shape_aspect_deriving_relationship
(*)
shape_aspect_relationship
*)
    ENTITY shape_aspect_relationship;
        name                : label;
        description          : OPTIONAL text;
        relating_shape_aspect : shape_aspect;
        related_shape_aspect  : shape_aspect;
    DERIVE
        id : identifier := get_id_value(SELF);
    WHERE
        wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
            'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
    END_ENTITY; -- shape_aspect_relationship
(*)
shape_defining_relationship
*)
    ENTITY shape_defining_relationship
        SUBTYPE OF (shape_aspect_relationship);
    END_ENTITY; -- shape_defining_relationship
(*)
shape_definition_representation
*)
    ENTITY shape_definition_representation
        SUBTYPE OF (property_definition_representation);
    WHERE
        wr1: ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(SELF.
            definition)) OR ('CAST_PARTS_SCHEMA.SHAPE_DEFINITION' IN
            TYPEOF(SELF.definition.definition));
        wr2: 'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION' IN TYPEOF(SELF.
            used_representation);
    END_ENTITY; -- shape_definition_representation
(*)
shape_dimension
*)
    ENTITY shape_dimension
        SUBTYPE OF (property_definition);
    WHERE

    (*
        shape_dimension to dimensional_tolerance(as dimensionals)
        shape_dimension to production_die_mould (as tooling)
        shape_dimension to production_core_box (as tooling)
        shape_dimension to production_pattern_definition (as tooling)
        shape_dimension to production_investment_cast_mould (as tooling)
        shape_dimension to production_die_cast_mould (as tooling)
    *)

    wr1: (NOT (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
        'CAST_PARTS_SCHEMA.'+

```

```

'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
( ('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE'
IN TYPEOF(pdr.related_property_definition)) AND
(NOT (SIZEOF(QUERY( it <* USEDIN(pdr.related_property_definition,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') |
(SIZEOF(['CAST_PARTS_SCHEMA.PRODUCTION_DIE_MOULD',
'CAST_PARTS_SCHEMA.PRODUCTION_CORE_BOX',
'CAST_PARTS_SCHEMA.PRODUCTION_PATTERN_DEFINITION',
'CAST_PARTS_SCHEMA.PRODUCTION_DIE_CAST_MOULD',
'CAST_PARTS_SCHEMA.PRODUCTION_INVESTMENT_CAST_MOULD']
* TYPEOF(it)) = 1)
)) =1 )) )) ) =0 ));

```

```
END_ENTITY; -- shape_dimension
```

```
(*
```

### **shape\_dimension\_representation**

```
*)
```

```

ENTITY shape_dimension_representation
SUBTYPE OF (shape_representation);
WHERE
  wr1: SIZEOF(QUERY ( temp <* SELF.items | (NOT (
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
    temp))) )) = 0;
  wr2: SIZEOF(SELF.items) <= 2;
  wr3: SIZEOF(QUERY ( pos_mri <* QUERY ( real_mri <* SELF.items | (
    'REAL' IN TYPEOF(real_mri\measure_with_unit.value_component)) )
    | (NOT (pos_mri\measure_with_unit.value_component > 0)) ))
    = 0;

```

```
END_ENTITY; -- shape_dimension_representation
```

```
(*
```

### **shape\_representation**

```
*)
```

```

ENTITY shape_representation
SUBTYPE OF (representation);
END_ENTITY; -- shape_representation

```

```
(*
```

### **shape\_representation\_with\_parameters**

```
*)
```

```

ENTITY shape_representation_with_parameters
SUBTYPE OF (shape_representation);
WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
    'CAST_PARTS_SCHEMA.PLACEMENT',
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'] *
    TYPEOF(it)) = 1)) )) = 0;

```

```
END_ENTITY; -- shape_representation_with_parameters
```

```
(*
```

### **shot\_sleeve**

```
*)
```

```

ENTITY shot_sleeve
SUBTYPE OF (die_design_feature);

```



```

WHERE

(* ----- shot_sleeve to item_size (as shot_sleeve_size) ----- *)

    wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='shot sleeve occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'shot sleeve shape usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relatng_shape_aspect))
    )) =1 ))) >=1;

END_ENTITY; -- shot_sleeve
(*
si_unit
*)
ENTITY si_unit
  SUBTYPE OF (named_unit);
  prefix : OPTIONAL si_prefix;
  name   : si_unit_name;
  DERIVE
    SELF\named_unit.dimensions : dimensional_exponents :=
                                dimensions_for_si_unit(name);
END_ENTITY; -- si_unit
(*
simulated_property
*)
ENTITY simulated_property
  SUBTYPE OF (representation);
  WHERE
  (*   property_value   *)

WR1: SIZEOF(QUERY( it <* SELF.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND
  ('CAST_PARTS_SCHEMA.PARAMETER_VALUE'
  IN TYPEOF(it\measure_with_unit.value_component)) AND
  (it.name = 'property value')))=1;

END_ENTITY; -- simulated_property
(*
simulation_input
*)
ENTITY simulation_input

  SUBTYPE OF (requirement_for_action_resource);
  WHERE
  (* simulation_input to CAST_PARTS_SCHEMA (as setup_file)   *)

wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,

```

## ISO 10303-223:2008 (E)

```
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
('CAST_PARTS_SCHEMA.DOCUMENT' IN TYPEOF
(adr.assigned_document)))=1;

(* simulation_input to condition_or_assumption(as condition) *)

wr2: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.CONDITION_OR_ASSUMPTION' IN TYPEOF(ap) ) AND
  (ap.name='condition'))
)) >= 0;

(* simulation_input to condition_or_assumption(as assumption) *)

wr3: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.CONDITION_OR_ASSUMPTION' IN TYPEOF(ap) ) AND
  (ap.name='assumption'))
)) >= 0;

(* to integration_interval(as integration_time_specification) *)

wr4: SIZEOF(QUERY ( ap <* SELF.OPERATIONS |
  (('CAST_PARTS_SCHEMA.INTEGRATION_INTERVAL' IN TYPEOF(ap) ) AND
  (ap.name='assumption'))
)) >= 1;

(* simulation_input to numeric_parameter(as input_parameter) *)

wr5: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
  (NOT(SIZEOF(QUERY ( ap <* USEDIN(rp,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
  ((SIZEOF(QUERY(ri <* ap.representation.items |
  (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(ri)) AND
  (ri.name = 'input parameter')))) >=1 )))) =0 )))) =0 ));

(* simulation_input to property_relationship(as input_parameter) *)

wr6: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
  (SIZEOF(QUERY ( ap <* USEDIN(rp,
  'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
  (('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN
  TYPEOF(ap.representation) ) )) >=1 )))) =0));

(* simulation_input to simulation_input_region (as region) *)

wr7: SIZEOF(QUERY ( r <* SELF.resources |
  'CAST_PARTS_SCHEMA.SIMULATION_INPUT_REGION'
  IN TYPEOF(r)) ) >= 1;

END_ENTITY; -- simulation_input
(*
```

### **simulation\_input\_region**

1690

© ISO 2008—All rights reserved

```

*)
ENTITY simulation_input_region
  SUBTYPE OF (action_resource);
  WHERE
  (* simulation_input_region to boundary_conditon(as boundary_condition) *)

  wr1: SIZEOF(QUERY( am <* SELF.usage |
    ('CAST_PARTS_SCHEMA.BOUNDARY_CONDITION' IN TYPEOF(am))))>=0;

  (* simulation_input_region to boundary_conditon(as boundary_condition) *)

  wr2: SIZEOF(QUERY( am <* SELF.usage |
    ('CAST_PARTS_SCHEMA.MESHING_CONDITION' IN TYPEOF(am))))>=0;

  (* simulation_input_region to numeric_parameter(as region_property)*)

  wr3: (NOT (SIZEOF(QUERY( am <* SELF.usage |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
    (NOT(SIZEOF(QUERY( ama <* USEDIN(am,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (NOT (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY(mri <* pd.used_representation.items |
    ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(mri)) = 2)AND
    (mri.name = 'region property'))
    )) >=1 ))) =0 ))))) =0 ))))) =0 ))))) =0)));

  (* simulation_input_region to property_relationship(as region_property) *)

  wr4: (NOT (SIZEOF(QUERY( am <* SELF.usage |
    (('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
    (NOT(SIZEOF(QUERY( ama <* USEDIN(am,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (NOT (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY(mri <* pd.used_representation.items |
    ('CAST_PARTS_SCHEMA.PROPERTY_RELATIONSHIP' IN TYPEOF(mri))))=1)
    )) >=1 ))))) =0 ))))) =0 ))))) =0)));

  (* simulation_input_region to shape_aspect (as area_to) *)

  wr5: (NOT (SIZEOF(QUERY( am <* SELF.usage |

```

```
(('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
(NOT(SIZEOF(QUERY( ama <* USEDIN(am,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama))
AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
)) =1 )))) =0 )))) =0));
```

(\* simulation\_input\_region to part\_version (as associated\_part) \*)

```
wr6: (NOT (SIZEOF(QUERY( am <* SELF.usage |
(('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(am)) AND
(NOT(SIZEOF(QUERY( ama <* USEDIN(am,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (eds))
)) >=0 )))) =0 )))) =0));
```

END\_ENTITY; -- simulation\_input\_region

(\*

## simulation\_output

\*)

```
ENTITY simulation_output
  SUBTYPE OF (versioned_action_request);
  WHERE
```

(\* simulation\_output to defect\_prediction(as predicted\_defect) \*)

```
wr1: SIZEOF(QUERY ( ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') |
('CAST_PARTS_SCHEMA.DEFECT_PREDICTION' IN TYPEOF(ars.method))))>=0;
```

(\* simulation\_output to simulation\_result(as result) \*)

```
wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_REQUEST_SOLUTION.REQUEST') | (
'CAST_PARTS_SCHEMA.SIMULATION_RESULT' IN TYPEOF(ars)) )) >= 0;
```

END\_ENTITY; -- simulation\_output

(\*

## simulation\_output\_region

\*)

```
ENTITY simulation_output_region
  SUBTYPE OF (action_property);
  WHERE
```

(\* simulation\_output\_region to simulation\_input\_region(as belongs\_to) \*)

```
wr1: ('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(SELF.definition)) AND
(SIZEOF(QUERY ( am <* USEDIN(SELF.definition,
'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') | (
'CAST_PARTS_SCHEMA.SIMULATION_INPUT_REGION' IN TYPEOF(am)) ))
```

```

= 1);

(* simulation_output_region to simulation_unit_state(as history) *)

wr2: SIZEOF(QUERY ( am <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.'+
  'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
  ('CAST_PARTS_SCHEMA.SIMULATION_UNIT_STATE'
  IN TYPEOF(am.related_action_property)
  ))>=1;

(* simulation_output_region to shape_element(as region_shape) *)

wr3: ('CAST_PARTS_SCHEMA.ACTION_METHOD' IN TYPEOF(SELF.definition)) AND
(NOT(SIZEOF(QUERY(am <* USEDIN(SELF.definition,
  'CAST_PARTS_SCHEMA.'+
  'ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(am)) AND
  (SIZEOF(QUERY(eds <* am.items |
  ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
  )) =1 ))))=0));

END_ENTITY; -- simulation_output_region
(*)
simulation_process
*)
ENTITY simulation_process
  SUBTYPE OF (casting_activity);
  WHERE
  (* --- simulation_process to design_part (as part_geometry) ----- *)
  (* --- simulation_process to design_part (as part_geometry) ----- *)
  (* --- simulation_process to design_part (as cast_part_with_rigging) -- *)

wr1: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(sar)) AND
  (NOT(SIZEOF(QUERY ( ip <* sar.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.CAST_PART',
  'CAST_PARTS_SCHEMA.CAST_PART_WITH_RIGGING',
  'CAST_PARTS_SCHEMA.DESIGN_PART'] * TYPEOF(ip)) =1))
  ) =0 ))))>=0);

(* -- to customer_casting_requirement (as quality_requirements) -- *)

wr2: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
  (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
  IN TYPEOF(ama)) AND
  (SIZEOF(QUERY(eds <* ama.items |
  (('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT' IN TYPEOF (eds))
  ))) >=1 ))))=0));

```

```

(* -- simulation_process to simulation_run (as run_simulation) -- *)

wr3: (SIZEOF(QUERY(ar <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION.CHOSEN_METHOD') |
('CAST_PARTS_SCHEMA.SIMULATION_RUN' IN TYPEOF(ar) )) )=1);

(* ---- to tooling(as tooling_for_simulation) ----- *)

wr4:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF (eds)) AND
(SIZEOF(['CAST_PARTS_SCHEMA.DIE_CAST_TOOLING',
'CAST_PARTS_SCHEMA.DIE_MOULD_TOOLING',
'CAST_PARTS_SCHEMA.SAND_CASTING_TOOLING',
'CAST_PARTS_SCHEMA.CORE_BOX_TOOLING',
'CAST_PARTS_SCHEMA.INVESTMENT_CASTING_TOOLING'] * TYPEOF(eds)) =1)
))) =1 ))) =0));

END_ENTITY; -- simulation_process
(*)
simulation_result
*)
ENTITY simulation_result
SUBTYPE OF (action_request_solution);
WHERE
(* begin_time *)

wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
((adr.role.name='begin time') AND
('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF
(adr.assigned_time))))=1;

(* end_time *)

wr2: SIZEOF(QUERY (adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
((adr.role.name='end time') AND
('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF
(adr.assigned_time))))=1;

(* simulation_result to simulation_output_region (as region) *)

wr3: (NOT (SIZEOF(QUERY( am <* USEDIN(SELF.method,
'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
(SIZEOF(QUERY( sor <* USEDIN(am.related_method,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
('CAST_PARTS_SCHEMA.SIMULATION_OUTPUT_REGION' IN TYPEOF(sor))
)) >=1 )))=0));

```

```

(* imulation_result to illustration (as illustration) *)

wr4: SIZEOF(QUERY (adr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
  ('CAST_PARTS_SCHEMA.ILLUSTRATION' IN TYPEOF
  (adr.assigned_document))))>=0;

END_ENTITY; -- simulation_result
(*)
simulation_run
*)
ENTITY simulation_run
  SUBTYPE OF (directed_action);
  WHERE
(* identification *)

wr1: SIZEOF(QUERY( arr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ID_ATTRIBUTE.IDENTIFIED_ITEM') |
  ('CAST_PARTS_SCHEMA.ID_ATTRIBUTE'
  IN TYPEOF(arr))))=1;

(* simulation_run to activity (as associated_process) *)

wr2: ('CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY'
  IN TYPEOF(SELF.chosen_method));

(* simulation_run to simulation_input (as input) *)

wr3: SIZEOF(QUERY( arr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
  ('CAST_PARTS_SCHEMA.SIMULATION_INPUT'
  IN TYPEOF(arr))))=1;

(* simulation_run to simulation_output (as output) *)

wr4: SIZEOF(QUERY ( r <* SELF.directive.requests |
  'CAST_PARTS_SCHEMA.SIMULATION_OUTPUT'
  IN TYPEOF(r)) = 1;

(* simulation_run to simulation_software (as software_used) *)

wr5: SIZEOF(QUERY( arr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('CAST_PARTS_SCHEMA.SIMULATION_SOFTWARE'
  IN TYPEOF(arr))))=1;

END_ENTITY; -- simulation_run
(*)
simulation_run_relationship
*)
ENTITY simulation_run_relationship
  SUBTYPE OF (action_method_relationship);

```

© ISO 2008 — All rights reserved

## ISO 10303-223:2008 (E)

```
WHERE
(* simulation_run_relationship to simulation_run (as related) *)

wr1: ('CAST_PARTS_SCHEMA.SIMULATION_RUN'
      IN TYPEOF(SELF.related_method));

(* simulation_run_relationship to simulation_run (as relating) *)

wr2: ('CAST_PARTS_SCHEMA.SIMULATION_RUN'
      IN TYPEOF(SELF.relying_method));

END_ENTITY; -- simulation_run_relationship
(*)
simulation_software
*)
ENTITY simulation_software
  SUBTYPE OF (action_resource);
  WHERE
  (* simulation_software to date (as release_date) *)

  wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
    ((adr.role.name='release date') AND
    ('CAST_PARTS_SCHEMA.DATE' IN TYPEOF
    (adr.assigned_date))))=1;

  END_ENTITY; -- simulation_software
(*)
simulation_unit
*)
ENTITY simulation_unit
  SUBTYPE OF (action_property);
  WHERE
  (* simulation_unit to location_element(as unit_location) *)

  wr1: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF.definition,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT' IN TYPEOF(ama))
    AND
    (NOT (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(eds)) AND
    (NOT(SIZEOF(QUERY(pd <* USEDIN(eds,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (SIZEOF(QUERY(mri <* pd.used_representation.items |
    ('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION'
    IN TYPEOF(mri))))=1)
    ))=1))))=0))))=0));

  (* simulation_unit to simulation_output_region(as belongs_to) *)

  wr2: SIZEOF(QUERY ( am <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.'+
    'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
```



```

('CAST_PARTS_SCHEMA.SIMULATION_OUTPUT_REGION'
IN TYPEOF(am.related_action_property)
))=1;

(* simulation_unit to simulation_unit_state(as history)*)

wr4: SIZEOF(QUERY ( am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.'+
'ACTION_PROPERTY_RELATIONSHIP.RELATING_ACTION_PROPERTY') |
('CAST_PARTS_SCHEMA.SIMULATION_UNIT_STATE'
IN TYPEOF(am.related_action_property)
))>=1;

(* simulation_unit to shape_element(as unit_shape) *)

wr5: (NOT(SIZEOF(QUERY( ama <* USEDIN(SELF.definition,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF (eds))
)) =1 )) )=0));

END_ENTITY; -- simulation_unit
(*
simulation_unit_state
*)
ENTITY simulation_unit_state
SUBTYPE OF (action_property);
WHERE
wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_TIME_ASSIGNMENT.ITEMS') |
((adr.role.name = 'evaluation time') AND
('CAST_PARTS_SCHEMA.LOCAL_TIME' IN TYPEOF(adr.assigned_time)))
)) = 1;

wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
'CAST_PARTS_SCHEMA.SIMULATED_PROPERTY'
IN TYPEOF(apr.representation)
))>=1;

END_ENTITY; -- simulation_unit_state
(*
single_activity_relationship
*)
ENTITY single_activity_relationship
SUBTYPE OF (sequential_method);
WHERE
wr2: 'CAST_PARTS_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
relating_method);
wr3: SIZEOF(['CAST_PARTS_SCHEMA.MACHINE_SETUP',

```

```

                'CAST_PARTS_SCHEMA.ANCILLARY_SETUP'] * TYPEOF(SELF.
                related_method)) = 1;
    END_ENTITY; -- single_activity_relationship
    (*
slide
    *)
    ENTITY slide
        SUBTYPE OF (die_design_feature);
        WHERE

    (* ***** slide_material ***** *)

    wr1: SIZEOF(QUERY ( co <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS') |
        (('CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION' IN TYPEOF(co)) AND
        (co.name IN
        ['wood','cast iron','urethane','aluminum','plastic']))) ) = 1;

    (* ----- slide to shape_element (as shape)----- *)

    wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
        ((sa.description='slide occurrence') AND
        (SIZEOF(QUERY (sar <* USEDIN(sa,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
        (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
        (sar.description = 'slide shape usage') AND
        ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sar.relying_shape_aspect))
        )) >=1 )))) >=1;

    (* ***** slide_activation_method ***** *)

    w3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
        (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) AND
        (SIZEOF(QUERY (it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
        IN TYPEOF(it)) AND (it.name = 'slide activation method')))) =1 ))))=1;

    END_ENTITY; -- slide
    (*

```

**slot**

```

    *)
    ENTITY slot
        SUBTYPE OF (feature_definition);
        WHERE
            wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
                SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'CAST_PARTS_SCHEMA.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'CAST_PARTS_SCHEMA.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(impl_rep.
used_representation.items) = 1)) ) = 0)) ) = 0;
wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'swept shape occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | (SIZEOF(['CAST_PARTS_SCHEMA.SQUARE_U_PROFILE',
'CAST_PARTS_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
'CAST_PARTS_SCHEMA.ROUNDED_U_PROFILE',
'CAST_PARTS_SCHEMA.VEE_PROFILE',
'CAST_PARTS_SCHEMA.TEE_PROFILE',
'CAST_PARTS_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
relating_shape_aspect)) = 1) )) = 1)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'course of travel occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((sar.name =
'course of travel') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar)))) ) | ('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relying_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'end condition occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | (((sar.description =
'slot end usage') AND (sar.name IN ['course of travel start',
'course of travel end'])) AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | ((('CAST_PARTS_SCHEMA.SLOT_END' IN TYPEOF(
fcr.relying_shape_aspect)) AND (fcr.relying_shape_aspect.
description IN ['open', 'radiused', 'flat', 'woodruff'])) AND (
'CAST_PARTS_SCHEMA.SLOT' IN TYPEOF(fcr.related_shape_aspect.
of_shape.definition))) )) = 1)) )) = 2)) )) = 0;
wr5: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (

```

```

'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'end condition occurrence') AND (SIZEOF(
QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP' +
'.RELATED_SHAPE_ASPECT') | ((sar.description =
'slot end usage') AND (sar.name IN ['course of travel start',
'course of travel end'])) AND (
'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
TYPEOF(sar))) ) | (('CAST_PARTS_SCHEMA.SLOT_END' IN TYPEOF(
fcr.relating_shape_aspect)) AND (fcr.relating_shape_aspect.
description IN ['loop'])) AND ('CAST_PARTS_SCHEMA.SLOT' IN
TYPEOF(fcr.related_shape_aspect.of_shape.definition))) =
1)) )) = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <*
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'course of travel occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'path feature component usage') AND ((sar.name =
'course of travel') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar)))) ) | (('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
relating_shape_aspect.description IN ['complex',
'complete circular'])) )) = 1)) )) = 1)) )) = 0);
wr6: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'maximum feature limit'))) ) >= 0;
END_ENTITY; -- slot
(*)

```

**slot\_end**

```

*)
ENTITY slot_end
SUBTYPE OF (shape_aspect);
WHERE
wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
SELF.of_shape.definition);
wr2 : SELF.description IN ['open','radiused','flat','woodruff',
'loop'];
wr3 : (NOT (SELF.description IN ['open','radiused','loop'])) OR (
SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd,'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
TYPEOF(pdr.used_representation))) = 0)) )) = 0);

```

```

wr4 : (NOT (SELF.description IN ['flat','woodruff'])) OR (SIZEOF(
  QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
wr5 : (NOT (SELF.description IN ['flat'])) OR (SIZEOF(QUERY ( pd <*
  USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(impl_rep
    .used_representation.items) = 2)) )) = 0)) )) = 0);
wr6 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
  USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'first radius')) )) = 1)) )) = 0)) )) =
    0);
wr7 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
  USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'second radius')) )) = 1)) )) = 0)) )) =
    0);
wr8 : (NOT (SELF.description = 'woodruff')) OR (SIZEOF(
  QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
    QUERY ( it <* impl_rep.used_representation.items | ((
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN TYPEOF(it))
    AND (it.name = 'radius')) )) = 1)) )) = 0)) )) = 0);

```

```

wr9 : (NOT (SELF.description IN ['woodruff'])) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
        TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(impl_rep
        .used_representation.items) = 1)) )) = 0)) )) = 0);
wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATING_SHAPE_ASPECT') | ((sar.description =
    'slot end usage') AND (sar.name IN [
    'course of travel start','course of travel end'])) AND (
    'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
    TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
    'end condition occurrence') AND ('CAST_PARTS_SCHEMA.SLOT'
    IN TYPEOF(fcr.related_shape_aspect.of_shape.definition)))
    AND ('CAST_PARTS_SCHEMA.SLOT_END' IN TYPEOF(fcr.
    relating_shape_aspect)))) ) >= 1;
END_ENTITY; -- slot_end
(*)
solid_angle_unit
*)
ENTITY solid_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
        wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\
        \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
        named_unit.dimensions.time_exponent = 0)) AND (SELF\
        named_unit.dimensions.electric_current_exponent = 0)) AND (
        SELF\named_unit.dimensions.
        thermodynamic_temperature_exponent = 0)) AND (SELF\
        named_unit.dimensions.amount_of_substance_exponent = 0)) AND
        (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- solid_angle_unit
(*)
solid_model
*)
ENTITY solid_model
    SUPERTYPE OF (manifold_solid_brep)
    SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- solid_model
(*)
special_inspection_requirement
*)
ENTITY special_inspection_requirement
    SUBTYPE OF (customer_casting_requirement);
    WHERE
    (* ----- inspection_description ----- *)

wr1: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND
(it.name = 'inspection description') AND
(it.description IN (['ultrasonic', 'other testing method',
                    'pressure test','x ray radiography',
                    'welding','liquid penetrant',
                    'magnetic particle'] ))
) )) =1 )))=0));

(* ----- other_testing_method ----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND
(it.name = 'other testing method')
) ))<=1 ))) = 0);

(* ----- performed_by_customer ----- *)

wr3: (NOT(SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
( SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND
(it.name = 'performed by customer') AND
(it.description IN (['TRUE','FALSE'] ))
) )) >=1 )))=0));

(* - to specification (as inspection_requirements)----- *)

wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
(adr.assigned_document.kind.product_data_type=
'inspection requirements') ))
>= 1;

END_ENTITY; -- special_inspection_requirement
(*
spherical_cap
*)
ENTITY spherical_cap
SUBTYPE OF (feature_definition);
WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +

```

```

        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 3)) )) = 0)) )) = 0;
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
        it)) = 2) AND (it.name = 'internal angle')) )) = 1)) )) = 0)) ))
        = 0;

```

END\_ENTITY; -- spherical\_cap

(\*

## spherical\_surface

\*)

```

ENTITY spherical_surface
  SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- spherical_surface

```

(\*

## sprue

\*)

```

ENTITY sprue
  SUBTYPE OF (gating_design_feature);
  WHERE

```

(\* --sprues to item\_size (as sprue\_dimensions) -- \*)

```

WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatng_shape_aspect))

```



```

    )) =1 )))) >=1;

(* **sprues to filter (as metal_flow_filter) * *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='sprue occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'filter reference usage') AND
  ('CAST_PARTS_SCHEMA.FILTER' IN TYPEOF(sar.relatinq_shape_aspect))
  )) <=1 )))) >=1;

(* ***** sprues to well (as sprue_well) ***** *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='sprue occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'well reference usage') AND
  ('CAST_PARTS_SCHEMA.WELL' IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

(* ----- sprue core ----- *)

wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELf) |
  (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
  TYPEOF(pdr.used_representation)) AND
  (SIZEOF(QUERY (it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM'
  IN TYPEOF(it)) AND (it.name = 'sprue core')))) =1 ))))=1;

(* --sprues to item_size (as pouring_cup) -- *)

WR5:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELf) |
  ((sa.description='sprue occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue pouring cup size usage') AND
  ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
  IN TYPEOF(sar.relatinq_shape_aspect))
  )) >=0 )))) >=1;

END_ENTITY; -- sprue
(*
sprue_and_runner_mould
*)
ENTITY sprue_and_runner_mould
  SUBTYPE OF (investment_design_feature);
  WHERE

```

```

(* --sprue_and_runner_mould to sprue ( as pouring_sprue) -- *)

wr1:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue and runner mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'sprue reference usage') AND
  ('CAST_PARTS_SCHEMA.SPRUE' IN TYPEOF(sar.relateing_shape_aspect))
  )) =1 )))) >=1;

(* -- sprue_and_runner_mould to runner (as gate_runner) -- *)

wr2: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue and runner mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'runner reference usage') AND
  ('CAST_PARTS_SCHEMA.RUNNER' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=1 )))) >=1;

(* -- sprue_and_runner_mould to ingate (as grigging_component) -- *)

wr3: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue and runner mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'ingate reference usage') AND
  ('CAST_PARTS_SCHEMA.INGATE' IN TYPEOF(sar.relateing_shape_aspect))
  )) >=1 )))) >=1;

(* -- sprue_and_runner_mould to ingate (as mould_assembly) -- *)

wr4: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
  ((sa.description='sprue and runner mould occurrence') AND
  (SIZEOF(QUERY (sar <* USEDIN(sa,
  'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
  (('CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar)) AND
  (sar.description = 'investment mould reference usage') AND
  ('CAST_PARTS_SCHEMA.INVESTMENT_MOULD'
  TYPEOF(sar.relateing_shape_aspect))
  )) =1 )))) >=1;
IN

END_ENTITY; -- sprue_and_runner_mould
(*
square_u_profile
*)
ENTITY square_u_profile
  SUBTYPE OF (shape_aspect);
  WHERE

```

```

wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
    SELF.of_shape.definition);
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) = 1)) ) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | ((NOT (SIZEOF(impl_rep.
    used_representation.items) >= 4)) AND (SIZEOF(impl_rep.
    used_representation.items) <= 7)) ) = 0)) ) = 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
    used_representation.items | ((((((srwp_i.name =
    'orientation') OR (srwp_i.name = 'width')) OR (srwp_i.name =
    'first angle')) OR (srwp_i.name = 'second angle')) OR (
    srwp_i.name = 'first radius')) OR (srwp_i.name =
    'second radius')) OR (srwp_i.name = 'profile limit')) OR (
    srwp_i.name = 'depth')) ) = SIZEOF(pdr.used_representation
    .items))) ) = 1)) ) = 1;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((
    'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
    'orientation')) ) = 1)) ) = 0)) ) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([

```

```

        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'width')) ) = 1)) ) = 0)) ) = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'first radius')) ) <= 1)) ) = 0)) )
    = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'second radius')) ) <= 1)) ) = 0)) )
    = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'first angle')) ) = 1)) )
    = 0)) ) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *

```

```

        TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
        = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.'
        + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
        pdr.used_representation)) AND (pdr.used_representation.name
        = 'profile limit')) )) <= 1)) )) = 0;
wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'depth')) )) <= 1)) )) = 0)) )) = 0;
END_ENTITY; -- square_u_profile

```

(\*

## **standard\_uncertainty**

\*)

```

ENTITY standard_uncertainty
  SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY; -- standard_uncertainty

```

(\*

## **step**

\*)

```

ENTITY step
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
        SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(impl_rep.
        used_representation.items) = 1)) )) = 0)) )) = 0;
    wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'course of travel occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =

```

```

        'path feature component usage') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | (('CAST_PARTS_SCHEMA.PATH_FEATURE_COMPONENT' IN
        TYPEOF(sdr.relying_shape_aspect)) AND (sdr.
        relating_shape_aspect.description = 'linear')) ) = 1)) ) =
        1)) ) = 0;
wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
        | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'removal boundary occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
        sar))) ) | ('CAST_PARTS_SCHEMA.VEE_PROFILE' IN TYPEOF(sdr.
        relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) | (('CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'maximum feature limit')) ) >= 0;
wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pds))
        AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
        'CAST_PARTS_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(csa))
        AND (csa.name = 'uncut area')) AND (SIZEOF(QUERY ( sar <*
        csa.component_relationships | ((
        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) AND (SIZEOF(['CAST_PARTS_SCHEMA.BOSS',
        'CAST_PARTS_SCHEMA.PROTRUSION'] * TYPEOF(sar.
        related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
END_ENTITY; -- step
(*)
stop_pin
*)
ENTITY stop_pin
  SUBTYPE OF (ejector_design_feature);
  WHERE
    wr1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='stop pin occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'stop pin size usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) = 1 )))) >=1;

END_ENTITY; -- stop_pin

```

(\*)

**straightness\_tolerance**

\*)

```

ENTITY straightness_tolerance
  SUBTYPE OF (geometric_tolerance);
  WHERE
    wr1: NOT (('CAST_PARTS_SCHEMA.' +
              'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- straightness_tolerance

```

(\*)

**surface**

\*)

```

ENTITY surface
  SUPERTYPE OF (ONEOF (elementary_surface, swept_surface, bounded_surface))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

```

(\*)

**surface\_curve**

\*)

```

ENTITY surface_curve
  SUBTYPE OF (curve);
  curve_3d : curve;
  associated_geometry : LIST [1:2] OF pcurve_or_surface;
  master_representation : preferred_surface_curve_representation;
  DERIVE
    basis_surface : SET [1:2] OF surface := get_basis_surface(SELF);
  WHERE
    wr1: curve_3d.dim = 3;
    wr2: ('CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(associated_geometry[1]))
        OR (master_representation <> pcurve_s1);
    wr3: ('CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(associated_geometry[2]))
        OR (master_representation <> pcurve_s2);
    wr4: NOT ('CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(curve_3d));
END_ENTITY; -- surface_curve

```

(\*)

**surface\_finish**

\*)

```

ENTITY surface_finish
  SUBTYPE OF (non_casting_activity);
END_ENTITY; -- surface_finish

```

(\*)

**surface\_of\_linear\_extrusion**

\*)

```

ENTITY surface_of_linear_extrusion
  SUBTYPE OF (swept_surface);
  extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

```

(\*)

**surface\_of\_revolution**

\*)

```

ENTITY surface_of_revolution

```

## ISO 10303-223:2008 (E)

```
    SUBTYPE OF (swept_surface);
    axis_position : axis1_placement;
    DERIVE
    axis_line : line := (dummy_gri || curve()) || line(axis_position.
        location,dummy_gri || vector(axis_position.z,1));
    END_ENTITY; -- surface_of_revolution
(*)
surface_profile_tolerance
*)
    ENTITY surface_profile_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
    wr1: (NOT (('CAST_PARTS_SCHEMA.' +
        'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF)))
        OR (SIZEOF(SELF\geometric_tolerance_with_datum_reference.
            datum_system) <= 3);
    END_ENTITY; -- surface_profile_tolerance
(*)
swept_surface
*)
    ENTITY swept_surface
    SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
    SUBTYPE OF (surface);
    swept_curve : curve;
    END_ENTITY; -- swept_surface
(*)
symmetric_shape_aspect
*)
    ENTITY symmetric_shape_aspect
    SUBTYPE OF (shape_aspect);
    INVERSE
    basis_relationships : SET [1:?] OF shape_aspect_relationship FOR
        relating_shape_aspect;
    WHERE
    wr1: SIZEOF(QUERY ( x <* SELF.basis_relationships | (
        'CAST_PARTS_SCHEMA.CENTRE_OF_SYMMETRY' IN TYPEOF(x.
            related_shape_aspect)) )) >= 1;
    END_ENTITY; -- symmetric_shape_aspect
(*)
symmetry_tolerance
*)
    ENTITY symmetry_tolerance
    SUBTYPE OF (geometric_tolerance_with_datum_reference);
    WHERE
    wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
        datum_system) <= 3;
    END_ENTITY; -- symmetry_tolerance
(*)
tangent
*)
    ENTITY tangent
```



```

SUBTYPE OF (derived_shape_aspect);
WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- tangent
(*)
taper
*)
ENTITY taper
SUBTYPE OF (shape_aspect);
WHERE
  wr1: 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(SELF
    .of_shape.definition);
  wr2: SELF.description IN ['angle taper','diameter taper',
    'directed taper'];
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) = 1)) ) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(impl_rep.
    used_representation.items) = 1)) ) = 0)) ) = 0;
  wr5: (NOT (SELF.description = 'angle taper')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
    it)) = 2) AND (it.name = 'taper angle')) ) = 1)) ) = 0)) ) =
    0);
  wr6: (NOT (SELF.description = 'diameter taper')) OR (SIZEOF(
    QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
    SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*

```

```

impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'final diameter')))) = 1)) )) = 0)) )
= 0);
wr7: (NOT (SELF.description = 'directed taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] * TYPEOF(
it)) = 2) AND (it.name = 'angle')) ) = 1)) )) = 0)) )) = 0);
wr8: (NOT (SELF.description = 'directed taper')) OR (SIZEOF(
QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
TYPEOF(pdr.used_representation)) AND (pdr.
used_representation.name = 'direction')) ) = 1)) )) = 0);
END_ENTITY; -- taper

```

(\*

## tee\_profile

\*)

```

ENTITY tee_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
      SELF.of_shape.definition);
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) = 1)) )) = 0;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
      (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'CAST_PARTS_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | ((NOT (SIZEOF(impl_rep.
      used_representation.items) >= 9)) AND (SIZEOF(impl_rep.
      used_representation.items) <= 10)) )) = 0)) )) = 0;
    wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
used_representation.items | ((((((((((srwp_i.name =
'orientation') OR (srwp_i.name = 'width')) OR (srwp_i.name =
'depth')) OR (srwp_i.name = 'cross bar width')) OR (
srwp_i.name = 'cross bar depth')) OR (srwp_i.name =
'first offset')) OR (srwp_i.name = 'second offset')) OR (
srwp_i.name = 'first angle')) OR (srwp_i.name =
'second angle')) OR (srwp_i.name = 'radius')) ) = SIZEOF(
pdr.used_representation.items)) ) = 1) ) = 1;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) ) = 1)) ) = 0)) ) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'width')) ) = 1)) ) = 0)) ) = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'depth')) ) = 1)) ) = 0)) ) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +

```

```

'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'cross bar width')) ) = 1)) ) = 0)) )
= 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'cross bar depth')) ) = 1)) ) = 0)) )
= 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'first offset')) ) = 1)) ) = 0)) ) =
0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'second offset')) ) = 1)) ) = 0)) )
= 0;
wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((

```

```

'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
= 0)) )) = 0;
wr13: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
= 0)) )) = 0;
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'radius')) )) <= 1)) )) = 0)) )) = 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.'
+ 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
pdr.used_representation)) AND (pdr.used_representation.name
= 'profile limit')) )) <= 1)) )) = 0;
END_ENTITY; -- tee_profile
(*)
thread
*)
ENTITY thread
  SUBTYPE OF (feature_definition);
  WHERE
    wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
        SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
            'CAST_PARTS_SCHEMA.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) AND (8 <= SIZEOF(pdr.
used_representation.items)) AND (SIZEOF(pdr.
used_representation.items) <= 11)) = 1) )) = 1;
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'major diameter')))) = 1)) = 0)) ))
= 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'minor diameter')))) <= 1)) = 0)) ))
= 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
= 2) AND (it.name = 'pitch diameter')))) <= 1)) = 0)) ))
= 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.RATIO_MEASURE_WITH_UNIT'] * TYPEOF(it))

```

```

= 2) AND (it.name = 'number of threads')) )) = 1)) )) = 0)) ))
= 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
= 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) )) =
0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'hand')) AND (it.description IN
['left','right'])) )) = 1)) )) = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) = 0)) ))
= 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,

```

```

    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | (((
    'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
    TYPEOF(it)) AND (it.name = 'thread side')) AND ((it.
    description = 'internal') OR (it.description = 'external')))) ))
    = 1)) )) = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'crest')) )) <= 1)) )) = 0)) )) = 0;
wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
    | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
    description = 'partial area occurrence') AND (SIZEOF(
    QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | ((sar.description =
    'applied area usage') AND (
    'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
    sar))) ) | ('CAST_PARTS_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
    relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr13: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
    'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
    | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
    QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
    'RELATED_SHAPE_ASPECT') | ((sar.description =
    'applied shape') AND (
    'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
    sar))) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
    relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +

```



```

        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
        'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'fit class 2')) ) <= 1)) ) = 0)) )
        = 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
        (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'CAST_PARTS_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
        'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
        = 2) AND (it.name = 'nominal size')) ) <= 1)) ) = 0)) )
        = 0;
END_ENTITY; -- thread
(*)
time_assignment
*)
ENTITY time_assignment
  ABSTRACT SUPERTYPE;
  assigned_time : local_time;
  role          : time_role;
END_ENTITY; -- time_assignment
(*)
time_measure_with_unit
*)
ENTITY time_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'CAST_PARTS_SCHEMA.TIME_UNIT' IN TYPEOF(SELF\measure_with_unit.
unit_component);
END_ENTITY; -- time_measure_with_unit
(*)
time_role
*)
ENTITY time_role;
  name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF, 'CAST_PARTS_SCHEMA.' +
        'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- time_role
(*)
time_unit
*)
ENTITY time_unit

```

## ISO 10303-223:2008 (E)

```
SUBTYPE OF (named_unit);
WHERE
  wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF\named_unit.dimensions.mass_exponent = 0)) AND (SELF\named_unit.dimensions.time_exponent = 1)) AND (SELF\named_unit.dimensions.electric_current_exponent = 0)) AND (SELF\named_unit.dimensions.thermodynamic_temperature_exponent = 0)) AND (SELF\named_unit.dimensions.amount_of_substance_exponent = 0)) AND (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
END_ENTITY; -- time_unit
(*)
tolerance_inspection
*)
ENTITY tolerance_inspection
  SUBTYPE OF (inspection_or_test_result);
WHERE
  (* --tolerance_inspection to numeric_parameter (as tolerance_result)-- *)

wr1: (NOT (SIZEOF(QUERY( pdr <* USEDIN(SELF,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  (SIZEOF(QUERY( it <* pdr.used_representation.items |
  (SIZEOF(['CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) = 2) AND
  (it.name = 'tolerance result') )) >=1)) ) = 0));

  (* --tolerance_inspection to tolerance_requirement (as requirements)- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.'+
  'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
  ('CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
  IN TYPEOF(pdr.related_property_definition))
  )) >=1);

END_ENTITY; -- tolerance_inspection
(*)
tolerance_requirement
*)
ENTITY tolerance_requirement
  SUBTYPE OF (customer_casting_requirement);
WHERE
  (* ----- allowance_value ----- *)

wr1: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
  'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ( NOT(SIZEOF(QUERY ( it <* pdr.used_representation.items |
  (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
  AND
  (it.name = 'dimension method') AND
  (it.description IN ['commercial practice',
  'specification data', 'drawing data'] )
  ) ))=1 )))) = 0);
```

```

(* ----- gaging_fixtures_available----- *)

wr2: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND
(it.name = 'gaging fixtures available') AND
(it.description IN ['TRUE','FALSE'] )
) ))=1 ))) = 0);

(* ----- to_dimensional_tolerance(as tolerance_based_on)----- *)

wr3: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it))
AND
(it.name = 'tolerance based one') AND
(it.description IN ['mismatch','dimensional','geometric'] )
) ))=1 ))) = 0);

(* --tolerance_requirement to checking_aid_tool(as gaging_requirement)-- *)

wr4: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.CHECKING_AID_TOOL'
IN TYPEOF(pdr.related_property_definition)
))>=0);

(* --- tolerance_requirement to mismatch_tolerance( as mismatch)-- *)

wr5: (SIZEOF(QUERY ( pdr <* USEDIN(SELF ,
'CAST_PARTS_SCHEMA.'+
'PROPERTY_DEFINITION_RELATIONSHIP.RELATING_PROPERTY_DEFINITION') |
('CAST_PARTS_SCHEMA.MISMATCH_TOLERANCE'
IN TYPEOF(pdr.related_property_definition)
))<=1);

(* - tolerance_requirement to specification (as dimension_specification)- *)

wr6: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
(adr.assigned_document.kind.product_data_type=
'dimension specification') )) <=1;

END_ENTITY; -- tolerance_requirement
(*
tolerance_value
*)
ENTITY tolerance_value;

```

```

        lower_bound : measure_with_unit;
        upper_bound : measure_with_unit;
    WHERE
        wr1: upper_bound.value_component > lower_bound.value_component;
        wr2: upper_bound.unit_component = lower_bound.unit_component;
    END_ENTITY; -- tolerance_value
(*)
tolerance_zone
*)
    ENTITY tolerance_zone
        SUBTYPE OF (shape_aspect);
        defining_tolerance : SET [1:?] OF geometric_tolerance;
        form                : tolerance_zone_form;
    END_ENTITY; -- tolerance_zone
(*)
tolerance_zone_definition
*)
    ENTITY tolerance_zone_definition
        SUPERTYPE OF (ONEOF (projected_zone_definition,runout_zone_definition));
        zone                : tolerance_zone;
        boundaries          : SET [1:?] OF shape_aspect;
    END_ENTITY; -- tolerance_zone_definition
(*)
tolerance_zone_form
*)
    ENTITY tolerance_zone_form;
        name : label;
    END_ENTITY; -- tolerance_zone_form
(*)
tool_verification
*)
    ENTITY tool_verification
        SUBTYPE OF (casting_method);
        WHERE
    (* -----tool_verification to core_assembly (as core_fit)----- *)

    wr1: (SIZEOF(QUERY(am <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
        ('CAST_PARTS_SCHEMA.CORE_ASSEMBLY' IN TYPEOF(am.related_method)))
        <=1);

    (* -----tool_verification to core_print (as verify_clearance)--*)

    wr2: (NOT (SIZEOF(QUERY(am <* USEDIN(SELF,
        'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
        (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
        IN TYPEOF(am)) AND
        (SIZEOF(QUERY( cp <* am.items |
        ('CAST_PARTS_SCHEMA.CORE_PRINT' IN TYPEOF(cp)))) >=1 ))))=0));

    (* --tool_verification to design_part (as based_on_part_design)- *)

```

```

wr3: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(am)) AND
(SIZEOF(QUERY( cp <* am.items |
('CAST_PARTS_SCHEMA.DESIGN_PART' IN TYPEOF(cp)))) =1 ))))=0));

(* -----tool_verification to shape_dimensions (as verify)----- *)

wr4: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(am)) AND
(SIZEOF(QUERY( cp <* am.items |
('CAST_PARTS_SCHEMA.SHAPE_DIMENSION' IN TYPEOF(cp)))) =1 ))))=0));

(* tool_verification to tolerance_requirement (as customer_requirement)--*)

wr5: (NOT(SIZEOF(QUERY(am <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(am)) AND
(SIZEOF(QUERY( cp <* am.items |
('CAST_PARTS_SCHEMA.TOLERANCE_REQUIREMENT'
IN TYPEOF(cp)))) >=1 ))))=0));

END_ENTITY; -- tool_verification
(*
tooling_process
*)
ENTITY tooling_process
SUBTYPE OF (casting_activity);
WHERE

(* ----- number_of_impressions ----- *)

wr1: (NOT(SIZEOF(QUERY(ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT(SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
('CAST_PARTS_SCHEMA.COUNT_MEASURE'
IN TYPEOF(it\measure_with_unit.value_component)) AND
(it.name = 'number of impressions')
))) =1 ))) =0 ))) =0));

(* ----- shrinkage_factor ----- *)

wr2: (NOT(SIZEOF(QUERY(ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') |
(NOT(SIZEOF(QUERY(apr <* USEDIN(ap,

```

ISO 10303-223:2008 (E)

```
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY(it <* apr.representation.items |
((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it)) =2) AND
(it.name = 'shrinkage factor')
))) =1 ))) =0 )))) =0));
```

(\* - process) - \*)

```
wr3: (NOT (SIZEOF(QUERY(ap <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_DEFINITION') |
(NOT (SIZEOF(QUERY(apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
(SIZEOF(QUERY(it <* apr.representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
IN TYPEOF(it)) AND
(it.name='process method') AND
(it.description IN [
'casting process',
'non permanent moulding process',
'permanent moulding process'] )
))) >=0 ))) =0 )) ) =0));
```

(\* ---- tooling\_proces to core (as cores\_to\_make) ----- \*)

```
wr4: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
('CAST_PARTS_SCHEMA.CORE' IN TYPEOF(eds.definition))
))) =1 ))))=0));
```

(\* ---permanent\_mould\_process to gating\_system (as required\_gating) --- \*)

```
wr5: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
IN TYPEOF(ama)) AND
(SIZEOF(QUERY(eds <* ama.items |
(('CAST_PARTS_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF (eds)) AND
('CAST_PARTS_SCHEMA.GATING_SYSTEM' IN TYPEOF(eds.definition))
))) =1 ))))=0));
```

(\* -- to customer\_casting\_requirement (as customer\_requirements) - \*)

```
wr6: (NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
(('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
```

```

    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.CUSTOMER_CASTING_REQUIREMENT' IN TYPEOF (eds))
    ))) >=1 ))))=0));

(* ---- permanent_mould_process to design_part (as part_geometry) --- *)

wr7:(NOT (SIZEOF(QUERY( ama <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD') |
    (('CAST_PARTS_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
    IN TYPEOF(ama)) AND
    (SIZEOF(QUERY(eds <* ama.items |
    (('CAST_PARTS_SCHEMA.DESIGN_PART' IN TYPEOF (eds))
    ))) =1 ))))=0));

(* ----- to tool_verification (as quality_assurance) ----- *)

wr8:(SIZEOF(QUERY(amr <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    ('CAST_PARTS_SCHEMA.TOOL_VERIFICATION' IN TYPEOF(amr.related_method)
    )))=1);

    END_ENTITY; -- tooling_process
(*
topological_representation_item
*)
    ENTITY topological_representation_item
        SUPERTYPE OF (ONEOF (vertex, edge, face_bound, face, connected_face_set,
            loop ANDOR path))
        SUBTYPE OF (representation_item);
    END_ENTITY; -- topological_representation_item
(*
toroidal_surface
*)
    ENTITY toroidal_surface
        SUBTYPE OF (elementary_surface);
        major_radius : positive_length_measure;
        minor_radius : positive_length_measure;
    END_ENTITY; -- toroidal_surface
(*
total_runout_tolerance
*)
    ENTITY total_runout_tolerance
        SUBTYPE OF (geometric_tolerance_with_datum_reference);
        WHERE
            wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
                datum_system) <= 2;
    END_ENTITY; -- total_runout_tolerance
(*
transition_feature
*)
    ENTITY transition_feature

```

```

SUPERTYPE OF (ONEOF (chamfer, edge_round, fillet))
SUBTYPE OF (shape_aspect);
WHERE
  wr1: SIZEOF(['CAST_PARTS_SCHEMA.PRODUCT_DEFINITION',
             'CAST_PARTS_SCHEMA.COMPOUND_FEATURE'] * TYPEOF(SELF.of_shape
             .definition)) = 1;
  wr2: SIZEOF(['CAST_PARTS_SCHEMA.CHAMFER',
             'CAST_PARTS_SCHEMA.EDGE_ROUND', 'CAST_PARTS_SCHEMA.FILLET'] *
             TYPEOF(SELF)) = 1;
END_ENTITY; -- transition_feature
(*)

```

## turned\_knurl

```

(*)
ENTITY turned_knurl
SUBTYPE OF (feature_definition);
WHERE
  wr1 : SELF\characterized_object.description IN ['diamond', 'diagonal',
          'straight'];
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
          SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
          'CAST_PARTS_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) AND ((6 <= SIZEOF(pdr.
          used_representation.items)) AND (SIZEOF(pdr.
          used_representation.items) <= 9))) ) ) = 1) ) = 1;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
          (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'CAST_PARTS_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'CAST_PARTS_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | (((
          'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
          it) AND ('CAST_PARTS_SCHEMA.COUNT_MEASURE' IN TYPEOF(it\
          measure_with_unit.value_component)) AND (it.name =
          'number of teeth')) ) <= 1)) ) = 0)) ) = 0;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
          (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'CAST_PARTS_SCHEMA.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'CAST_PARTS_SCHEMA.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
          'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
          = 2) AND (it.name = 'major diameter')) ) = 1)) ) = 0)) )
          = 0;

```



```

wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'nominal diameter')) ) = 1)) ) = 0)) )
    = 0;

wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.' + 'DEFINITION') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'tooth depth')) ) <= 1)) ) = 0)) ) =
    0;

wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'root fillet')) ) <= 1)) ) = 0)) ) =
    0;

wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] * TYPEOF(it))
    = 2) AND (it.name = 'diametral pitch')) ) = 1)) ) = 0)) )
    = 0;

wr9 : (NOT (SELF\characterized_object.description IN ['diamond',

```

```

    'diagonal'])) OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
    'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation))) | (NOT (SIZEOF(QUERY ( it <*
    impl_rep.used_representation.items | ((SIZEOF([
    'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
    'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'helix angle')) ) = 1)) )
    = 0)) ) = 0);
wr10: (NOT (SELF\characterized_object.description = 'diagonal')) OR
(SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation))) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.DESCRPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(it)) AND (it.name = 'helix hand')) ) = 1)) ) = 0)) )
= 0);
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
description = 'partial area occurrence') AND (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'applied area usage') AND (('CAST_PARTS_SCHEMA.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
'CAST_PARTS_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
relating_shape_aspect))) = 1)) ) = 1)) ) = 0;
wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
'CAST_PARTS_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(pd)) )
| (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
'RELATED_SHAPE_ASPECT') | ((sar.description =
'applied shape') AND (
'CAST_PARTS_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN TYPEOF(
sar))) ) | ('CAST_PARTS_SCHEMA.SHAPE_ASPECT' IN TYPEOF(sdr.
relating_shape_aspect))) = 1)) ) = 1)) ) = 0;
END_ENTITY; -- turned_knurl
(*

```

**type\_qualifier**

```

*)
  ENTITY type_qualifier;
    name : label;
  END_ENTITY; -- type_qualifier
(*)

```

**uncertainty\_measure\_with\_unit**

```

*)
  ENTITY uncertainty_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    name : label;
    description : OPTIONAL text;
  WHERE
    wr1: valid_measure_value(SELF\measure_with_unit.value_component);
  END_ENTITY; -- uncertainty_measure_with_unit
(*)

```

**uncertainty\_qualifier**

```

*)
  ENTITY uncertainty_qualifier
    SUPERTYPE OF (standard_uncertainty);
    measure_name : label;
    description : text;
  END_ENTITY; -- uncertainty_qualifier
(*)

```

**uniform\_curve**

```

*)
  ENTITY uniform_curve
    SUBTYPE OF (b_spline_curve);
  END_ENTITY; -- uniform_curve
(*)

```

**uniform\_surface**

```

*)
  ENTITY uniform_surface
    SUBTYPE OF (b_spline_surface);
  END_ENTITY; -- uniform_surface
(*)

```

**value\_range**

```

*)
  ENTITY value_range
    SUBTYPE OF (compound_representation_item);
  WHERE

    wr1: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
      ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
      ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
    ))=2;

    wr2: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
      ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
      (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
      AND (mri.name='lower limit'))
    ))=1;

```

```

wr3: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
    ('CAST_PARTS_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
    (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
    AND (mri.name='upper limit')) )=1;

wr4: SIZEOF(QUERY( i1 <* SELF.item_element |
    ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i1)) AND
    (SIZEOF ( QUERY (i2 <* SELF.item_element |
    ('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i2)) AND
    (i1 :<>: i2) AND
    (i1\measure_with_unit.unit_component :=:
    i2\measure_with_unit.unit_component)
    ) ) = 1 ))) = 2 ;
END_ENTITY; -- value_range

```

(\*

**value\_representation\_item**

\*)

```

ENTITY value_representation_item
  SUBTYPE OF (representation_item);
  value_component : measure_value;
WHERE
  wr1: SIZEOF(QUERY ( rep <* using_representations(SELF) | (NOT (
    'CAST_PARTS_SCHEMA.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN TYPEOF(
    rep.context_of_items))) ) ) = 0;
END_ENTITY; -- value_representation_item

```

(\*

**vector**

\*)

```

ENTITY vector
  SUBTYPE OF (geometric_representation_item);
  orientation : direction;
  magnitude : length_measure;
WHERE
  wr1: magnitude >= 0;
END_ENTITY; -- vector

```

(\*

**vee\_profile**

\*)

```

ENTITY vee_profile
  SUBTYPE OF (shape_aspect);
WHERE
  wr1 : 'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN TYPEOF(
    SELF.of_shape.definition);
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
    (SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
    'CAST_PARTS_SCHEMA.' +
    'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
    used_representation))) ) = 1)) ) = 0;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | ((NOT (SIZEOF(impl_rep.
used_representation.items) >= 3)) AND (SIZEOF(impl_rep.
used_representation.items) <= 6)) ) = 0)) ) = 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
used_representation.items | ((((((srwp_i.name =
'orientation') OR (srwp_i.name = 'profile angle')) OR (
srwp_i.name = 'tilt angle')) OR (srwp_i.name =
'profile radius')) OR (srwp_i.name = 'first length')) OR (
srwp_i.name = 'second length')) ) = SIZEOF(pdr.
used_representation.items))) ) = 1) ) = 1;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((
'CAST_PARTS_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.name =
'orientation')) ) = 1)) ) = 0)) ) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
it)) AND ('CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF(it\measure_with_unit.value_component))) AND (it.name
= 'profile radius')) ) <= 1)) ) = 0)) ) = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.

```

```

used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'profile angle')) ) = 1)) )
= 0)) ) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | ((SIZEOF([
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM',
'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
TYPEOF(it)) = 2) AND (it.name = 'tilt angle')) ) = 1)) )
= 0)) ) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( pdr <* USEDIN(pd, 'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN TYPEOF(
pdr.used_representation)) AND (pdr.used_representation.name
= 'profile limit')) ) <= 1)) ) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
it)) AND ('CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF(it\measure_with_unit.value_component))) AND (it.name
= 'first length')) ) <= 1)) ) = 0)) ) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (NOT
(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
'CAST_PARTS_SCHEMA.' +
'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
impl_rep.used_representation.items | (((
'CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(
it)) AND ('CAST_PARTS_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
TYPEOF(it\measure_with_unit.value_component))) AND (it.name
= 'second length')) ) <= 1)) ) = 0)) ) = 0;
END_ENTITY; -- vee_profile

```

```

(*)
vent
*)
  ENTITY vent
    SUBTYPE OF (gating_design_feature);
    WHERE
  (* ----- type of vent ----- *)

  wr1:SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS' IN
    TYPEOF(pdr.used_representation)) AND
    (SIZEOF(QUERY (it <* pdr.used_representation.items |
    (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND ((it.name = 'type of vent') AND
    ((it.description IN ['air','gas'] )) )) )) =1 )) ))>=1;

  (* ----- vent to item_size (as core_vent_size) ----- *)

  WR2:SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='vent occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) AND
    (sar.description = 'vent reference usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relateing_shape_aspect))
    )) =1 ))) >=1;

  (* ----- vent to location_element (as vent_placement)----- *)

  wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(SELF) |
    (('CAST_PARTS_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
    TYPEOF(pdr.used_representation)) AND
    (pdr.used_representation.name='location'))))=1;
  END_ENTITY; -- vent
  (*
versioned_action_request
  *)
  ENTITY versioned_action_request;
    id          : identifier;
    version     : label;
    purpose     : text;
    description : OPTIONAL text;
  END_ENTITY; -- versioned_action_request
  (*
vertex
  *)
  ENTITY vertex
    SUBTYPE OF (topological_representation_item);
  END_ENTITY; -- vertex
  (*

```

**vertex\_loop**

```

*)
  ENTITY vertex_loop
    SUBTYPE OF (loop);
    loop_vertex : vertex;
  END_ENTITY; -- vertex_loop
(*

```

**vertex\_point**

```

*)
  ENTITY vertex_point
    SUBTYPE OF (vertex, geometric_representation_item);
    vertex_geometry : point;
  END_ENTITY; -- vertex_point
(*

```

**view\_reference**

```

*)
  ENTITY view_reference
    SUBTYPE OF (document_usage_constraint);
  WHERE
    wr1: NOT (SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF.source,
      'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT' ) |
      ('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN TYPEOF(dr)) )
      | (SIZEOF(QUERY ( d <* adr.items | (
      'CAST_PARTS_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) )
      >= 0) )) = 0);
    wr2: 'CAST_PARTS_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(SELF.source);
    wr3: SELF.subject_element IN ['sheet', 'view', 'zone'];
  END_ENTITY; -- view_reference
(*

```

**well**

```

*)
  ENTITY well
    SUBTYPE OF (gating_design_feature);
  WHERE

  (* ----- well to item_size (as well_dimensions) ----- *)

  WR1: SIZEOF(QUERY ( sa <* get_property_definition_shape_aspect(SELF) |
    ((sa.description='well occurrence') AND
    (SIZEOF(QUERY (sar <* USEDIN(sa,
    'CAST_PARTS_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT' ) |
    (('CAST_PARTS_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN TYPEOF(sar)) AND
    (sar.description = 'well size usage') AND
    ('CAST_PARTS_SCHEMA.CASTING_FEATURE_SIZE'
    IN TYPEOF(sar.relying_shape_aspect))
    )) = 1 )))) >= 1;

  (* ----- well to parting_surface (as drag_surface) ----- *)

  wr2: SIZEOF(QUERY( pdr <* get_property_definition_representations(SELF) |
    ('CAST_PARTS_SCHEMA.PARTING_SURFACE'
    IN TYPEOF(pdr.used_representation)) )) = 1;

```



```

END_ENTITY; -- well
(*)
application_context_requires_ap_definition
*)
RULE application_context_requires_ap_definition FOR (application_context,
    application_protocol_definition);

WHERE
    wr1: SIZEOF(QUERY ( ac <* application_context | (NOT (SIZEOF(
        QUERY ( apd <* application_protocol_definition | ((ac :=: apd.
            application) AND (apd.
                application_interpreted_model_schema_name =
                    'cast parts schema')) )) = 1)) )) = 0;

END_RULE; -- application_context_requires_ap_definition
(*)
casting_feature_life_cycle
*)
RULE casting_feature_life_cycle FOR (casting_instanced_feature);

WHERE
    wr1: SIZEOF(QUERY ( mf <* casting_instanced_feature | (NOT (mf.
        of_shape.definition.frame_of_reference.life_cycle_stage =
            'manufacturing planning')) )) = 0;

END_RULE; -- casting_feature_life_cycle
(*)
chamfer_offset_requires_faces
*)
RULE chamfer_offset_requires_faces FOR (chamfer_offset,
    property_definition_representation);

WHERE

    WR1: SIZEOF(QUERY ( co <* chamfer_offset |
        (( (co.description = 'first offset')) AND
            (SIZEOF(QUERY ( pd <* USEDIN(co,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
                (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                    'CAST_PARTS_SCHEMA.' +
                    'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
                    (('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
                    IN TYPEOF(pdr.used_representation)) AND
                    (pdr.used_representation.name = 'first chamfer face')) )) = 1)) ))
            = 1)) ))=0;

    WR2: SIZEOF(QUERY ( co <* chamfer_offset |
        (( (co.description = 'second offset')) AND
            (SIZEOF(QUERY ( pd <* USEDIN(co,
                'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
                (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,

```

```

        'CAST_PARTS_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND
        (pdr.used_representation.name = 'second chamfer face')) )) = 1)) ))
        = 1)) ))=0;
    END_RULE; -- chamfer_offset_requires_faces
    (*
chamfer_requires_faces
    *)
    RULE chamfer_requires_faces FOR (chamfer,
        property_definition_representation);

    WHERE

        wr1: SIZEOF(QUERY ( er <* chamfer |
            (SIZEOF(QUERY (pd <* USEDIN (er,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
            NOT (SIZEOF(QUERY (pdr <* USEDIN (pd,
            'CAST_PARTS_SCHEMA.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            ('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
            IN TYPEOF (pdr.used_representation)) AND
            (pdr.used_representation.name = 'chamfer face')))=1)))=1)))=0;

        END_RULE; -- chamfer_requires_faces
    (*
compatible_dimension
    *)
    RULE compatible_dimension FOR (cartesian_point, direction,
        representation_context, geometric_representation_context);

    WHERE

        wr1: SIZEOF(QUERY ( x <* cartesian_point | (SIZEOF(QUERY ( y <*
            geometric_representation_context | (item_in_context(x,y) AND (
            HIINDEX(x.coordinates) <> y.coordinate_space_dimension)) )) >
            0) )) = 0;

        wr2: SIZEOF(QUERY ( x <* direction | (SIZEOF(QUERY ( y <*
            geometric_representation_context | (item_in_context(x,y) AND (
            HIINDEX(x.direction_ratios) <> y.coordinate_space_dimension)) ))
            > 0) )) = 0;

        END_RULE; -- compatible_dimension
    (*
composition_element_requires_name
    *)
    RULE composition_element_requires_name FOR (
        product_definition,product_definition_formation);

    WHERE

        wr1: SIZEOF(QUERY ( pd <* product_definition |

```

```

((pd.frame_of_reference.name='material composition definition') AND
(NOT((SIZEOF(QUERY( pdf <* product_definition_formation |
(pd.formation ::= pdf) AND
(pdf.of_product.name IN [
'Actinium','Aluminum','Americium','Antimony','Argon','Arsenic','Astatine',
'Barium','Berkelium','Beryllium','Bismuth','Bohrium','Boron','Bromine',
' Cadmium','Caesium','Calcium','Californium','Carbon','Cerium','Chlorine',
'Chromium','Cobalt','Copper','Curium','Darmstadtium','Dubnium','Dysprosium',
'Einsteinium','Erbium','Europium','Fermium','Fluorine','Francium',
'Gadolinium','Gallium','Germanium','Gold','Hafnium','Hassium','Helium',
'Holmium','Hydrogen','Indium','Iodine','Iridium','Iron','Krypton',
'Lanthanum','Lawrencium','Lead','Lithium','Lutetium','Magnesium',
'Manganese','Meitnerium','Mendelevium','Mercury','Molybdenum','Neodymium',
'Neon','Neptunium','Nickel','Niobium','Nitrogen','Nobelium','Osmium',
'Oxygen','Palladium','Phosphorus','Platinum','Plutonium','Polonium',
'Potassium','Praseodymium','Promethium','Protactinium','Radium','Radon',
'Rhenium','Rhodium','Roentgenium','Rubidium','Ruthenium','Rutherfordium',
'Samarium','Scandium','Seaborgium','Selenium','Silicon','Silver','Sodium',
'Strontium','Sulfur','Tantalum','Technetium','Tellurium','Terbium',
'Thallium','Thorium','Thulium','Tin','Titanium','Tungsten','Ununbium',
'Ununhexium','Ununoctium','Ununpentium','Ununquadium','Ununtrium',
'Uranium','Vanadium','Xenon','Ytterbium','Yttrium','Zinc','Zirconium']
)) )) =1 )) )) =0;

```

```
END_RULE; --composition_element_requires_name
```

```
(*
```

## customer\_order\_rules

```
*)
```

```
RULE customer_order_rules FOR (directed_action);
```

```
WHERE
```

```

wr1 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,
'CAST_PARTS_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') | (ada.
role.name = 'delivery date') )) = 1))) )) = 0;
wr2 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ap <* USEDIN(da,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') | (NOT (
SIZEOF(QUERY ( apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
| (NOT (SIZEOF(QUERY ( rep <* apr.representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(rep)) AND (rep.name = 'production status')) AND (rep.
description IN ['current','experimental','prototype','new'])) ))
= 1)) )) = 0)) )) = 0))) )) = 0;
wr3 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( fr <* da.directive
.requests | ('CAST_PARTS_SCHEMA.FIRST_ARTICLE' IN TYPEOF(fr)) ))
>= 1))) )) = 0;
wr4 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,
'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATED_ACTION') | (
'CAST_PARTS_SCHEMA.INSPECTION_PLAN' IN TYPEOF(ada.

```

```

relating_action)) )) = 1))) )) = 0;
wr5 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( fr <* da.directive
.requests | ('CAST_PARTS_SCHEMA.REQUEST_FOR_QUOTATION' IN
TYPEOF(fr)) )) <= 1))) )) = 0;
wr6 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,
'CAST_PARTS_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION') | (
'CAST_PARTS_SCHEMA.ORDERED_PART' IN TYPEOF(ada)) )) = 1))) ))
= 0;
wr7 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ap <* USEDIN(da,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION') | (NOT (
SIZEOF(QUERY ( apr <* USEDIN(ap,
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
| (NOT (SIZEOF(QUERY ( rep <* apr.representation.items | (((
'CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN
TYPEOF(rep)) AND (rep.name = 'casting method')) AND (rep.
description IN ['core presentation',
'core setting',
'die assembly',
'die handling',
'die temperature control',
'furnace loading',
'gate system removal',
'grinding',
'inspection',
'ladle loading',
'loading injection system',
'melt raw material',
'mould closing',
'mould making process',
'package and shipping',
'part routing',
'pattern handling',
'tool handling',
'pouring process',
'refractor removal',
'removal from die',
'shakeout',
'slurry process',
'sand removal',
'wax removal',
'shell',
'green sand',
'no bake',
'lost form',
'vacuum',
'static',
'tilt',
'low pressure']))) )) = 1)) )) = 0)) )) = 0))) )) = 0;
wr8 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,

```

```

'CAST_PARTS_SCHEMA.APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT.ITEMS')
  | (ada.role.name = 'customer') )) = 1))) )) = 0;
wr9 : SIZEOF(QUERY ( da <* directed_action | ((da.name =
  'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,
  'CAST_PARTS_SCHEMA.ACTION_RELATIONSHIP.RELATED_ACTION') | (
  ada.relatng_action.name = 'project order') )) = 1))) )) = 0;
wr10: SIZEOF(QUERY ( da <* directed_action | ((da.name =
  'customer order') AND (NOT (SIZEOF(QUERY ( ada <* USEDIN(da,
  'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') | (ada.
  assigned_document\document.kind.product_data_type =
  'simulation report') )) = 1))) )) = 0;

```

```
END_RULE; -- customer_order_rules
```

```
(*
```

### **dependent\_instantiable\_approval\_status**

```
*)
```

```
RULE dependent_instantiable_approval_status FOR (approval_status);
```

```
WHERE
```

```
wr1: SIZEOF(QUERY ( ast <* approval_status | (NOT (SIZEOF(USEDIN(ast,
  '')) >= 1)) )) = 0;
```

```
END_RULE; -- dependent_instantiable_approval_status
```

```
(*
```

### **dependent\_instantiable\_date**

```
*)
```

```
RULE dependent_instantiable_date FOR (date);
```

```
WHERE
```

```
wr1: SIZEOF(QUERY ( dt <* date | (NOT (SIZEOF(USEDIN(dt, '')) >= 1)) ))
= 0;
```

```
END_RULE; -- dependent_instantiable_date
```

```
(*
```

### **dependent\_instantiable\_named\_unit**

```
*)
```

```
RULE dependent_instantiable_named_unit FOR (named_unit);
```

```
WHERE
```

```
wr1: SIZEOF(QUERY ( nu <* named_unit | (NOT (SIZEOF(USEDIN(nu, '')) >=
  1)) )) = 0;
```

```
END_RULE; -- dependent_instantiable_named_unit
```

```
(*
```

### **dependent\_instantiable\_precision\_qualifier**

```
*)
```

```
RULE dependent_instantiable_precision_qualifier FOR (precision_qualifier);
```

```
WHERE
```

```
wr1: SIZEOF(QUERY ( pq <* precision_qualifier | (NOT (SIZEOF(USEDIN(pq,
  '')) >= 1)) )) = 0;
```

ISO 10303-223:2008 (E)

```
END_RULE; -- dependent_instantiable_precision_qualifier
(*)
dependent_instantiable_security_classification_level
*)
RULE dependent_instantiable_security_classification_level FOR (
    security_classification_level);

WHERE
    wr1: SIZEOF(QUERY ( scl <* security_classification_level | (NOT (
        SIZEOF(USEDIN(scl, '')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_security_classification_level
(*)
dependent_instantiable_shape_representation
*)
RULE dependent_instantiable_shape_representation FOR (
    shape_representation);

WHERE
    wr1: SIZEOF(QUERY ( sr <* shape_representation | (NOT (SIZEOF(USEDIN(
        sr, '')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_shape_representation
(*)
dependent_instantiable_type_qualifier
*)
RULE dependent_instantiable_type_qualifier FOR (type_qualifier);

WHERE
    wr1: SIZEOF(QUERY ( tq <* type_qualifier | (NOT (SIZEOF(USEDIN(tq, ''))
        >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_type_qualifier
(*)
dependent_instantiable_uncertainty_qualifier
*)
RULE dependent_instantiable_uncertainty_qualifier FOR (
    uncertainty_qualifier);

WHERE
    wr1: SIZEOF(QUERY ( uq <* uncertainty_qualifier | (NOT (SIZEOF(USEDIN(
        uq, '')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_uncertainty_qualifier
(*)
document_file_rules
*)
RULE document_file_rules FOR (document_file);

WHERE
```

```

(* --quality_acceptance_report to reporting_requirement (as applied_to) - *)

wr1: ((SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'quality acceptance report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
    (('CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN TYPEOF(doc)) AND
    (SIZEOF(QUERY(rep <* doc.items |
      ('CAST_PARTS_SCHEMA.REPORTING_REQUIREMENT' IN TYPEOF(rep))
    ))>=1))))=0))))=0));

(* ----- Product_quality_report action_required ----- *)

wr2: (NOT(SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT(SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        (it.name = 'action required')) =1 ))) =0 )))) =0 )))) =0));

(* ----- Product_quality_report number_failed ----- *)

wr3: (NOT (SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    (NOT(SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (SIZEOF(QUERY( it <* pdr.used_representation.items |
        (('CAST_PARTS_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
        ('CAST_PARTS_SCHEMA.COUNT_MEASURE'
        IN TYPEOF(it\measure_with_unit.value_component)) AND
        (it.name = 'number failed')
        ) ) =1 ))) =0 )))) =0 )))) =0));

(* Product_quality_report to inspection_or_test_result (as what_failed) -*)

wr4: ((SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
    ('CAST_PARTS_SCHEMA.INSPECTION_OR_TEST_RESULT' IN TYPEOF(pd))))=1
  ))))=0));

(* -- customer_design_deficiency_report list_of_results ----- *)

wr5: (NOT(SIZEOF(QUERY(doc <* document_file |
  ((doc.kind.product_data_type = 'product quality report') AND
  (NOT(SIZEOF(QUERY ( pd <* USEDIN(doc,

```

```

'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
(NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
(SIZEOF(QUERY( it <* pdr.used_representation.items |
(('CAST_PARTS_SCHEMA.DESRIPTIVE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
(it.name = 'list of results')))) =1 ))) =0 ))) =0 )))) =0));

(* --- customer_tool_design_report to tooling (as designed_tool ----- *)

wr6: ((SIZEOF(QUERY(doc <* document_file |
((doc.kind.product_data_type = 'customer tool design report') AND
(SIZEOF(QUERY ( pd <* USEDIN(doc,
'CAST_PARTS_SCHEMA.DOCUMENT_PRODUCT_ASSOCIATION.RELATING_DOCUMENT') |
(NOT ('CAST_PARTS_SCHEMA.CASTING_PRODUCT_DEFINITION'
IN TYPEOF(pd.related_product)
))))=1 ))))=0));

END_RULE; -- document_file__rules
(*
edge_round_requires_faces
*)
RULE edge_round_requires_faces FOR (edge_round,
property_definition_representation);

WHERE

wr1: SIZEOF(QUERY ( er <* edge_round |
(SIZEOF (QUERY (pd <* USEDIN (er,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'edge round face')))) =1 ))) =1 ))))=0;

wr2: SIZEOF(QUERY ( er <* edge_round |
(SIZEOF (QUERY (pd <* USEDIN (er,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
IN TYPEOF (pdr.used_representation)) AND
(pdr.used_representation.name = 'first face shape')))) =1 ))) =1 ))))=0;

wr3: SIZEOF(QUERY ( er <* edge_round |
(SIZEOF (QUERY (pd <* USEDIN (er,
'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
NOT (SIZEOF (QUERY (pdr <* USEDIN (pd,
'CAST_PARTS_SCHEMA.' +
'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |

```



```

('CAST_PARTS_SCHEMA.FACE_SHAPE_REPRESENTATION'
 IN TYPEOF (pdr.used_representation)) AND
 (pdr.used_representation.name = 'second face shape')) =1 ))) =1 )))=0;

```

```
END_RULE; -- edge_round_requires_faces
```

```
(*
```

### **externally\_defined\_class\_with\_known\_source\_requirement**

```
*)
```

```
RULE externally_defined_class_with_known_source_requirement FOR (
  externally_defined_class);
```

```
WHERE
```

```

wr1: SIZEOF(QUERY ( edc <* externally_defined_class | ((
  'CAST_PARTS_SCHEMA.KNOWN_SOURCE' IN TYPEOF(edc.source)) AND (
  NOT (SIZEOF(QUERY ( aoa <* USEDIN(edc.source,
  'CAST_PARTS_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') | (
  aoa.role.name = 'library_supplier') )) = 1))) )) <= 1;

```

```
END_RULE; -- externally_defined_class_with_known_source_requirement
```

```
(*
```

### **machining\_feature\_life\_cycle**

```
*)
```

```
RULE machining_feature_life_cycle FOR (instanced_feature);
```

```
WHERE
```

```

wr1: SIZEOF(QUERY ( mf <* instanced_feature | (NOT (mf.of_shape.
  definition.frame_of_reference.life_cycle_stage =
  'manufacturing_planning')) )) = 0;

```

```
END_RULE; -- machining_feature_life_cycle
```

```
(*
```

### **material\_is\_specified\_for\_part**

```
*)
```

```
RULE material_is_specified_for_part FOR (product_definition,
  make_from_usage_option);
```

```
WHERE
```

```

wr1: SIZEOF(QUERY ( nmpd <* QUERY ( pd <* product_definition |
  (SIZEOF(USEDIN(pd, 'CAST_PARTS_SCHEMA.MATERIAL_DESIGNATION.DEFINITIONS'))
  = 0) ) |
  (NOT (SIZEOF(QUERY ( mfuo <* make_from_usage_option |
  (nmpd ::= mfuo.relatering_product_definition) )) >= 1)) )) = 0;

```

```
END_RULE; -- material_is_specified_for_part
```

```
(*
```

### **part\_requires\_project\_order**

```
*)
```

```
RULE part_requires_project_order FOR (product_definition,
  applied_action_assignment);
```

```
WHERE
```

```
wr1: SIZEOF(QUERY ( pd <* product_definition |
```

## ISO 10303-223:2008 (E)

```
((pd.frame_of_reference.name='part definition') AND
(NOT (SIZEOF(QUERY ( fbppaa <* applied_action_assignment |
(pd.formation IN fbppaa.items) AND
(fbppaa.assigned_action.name = 'project order')) )) =1 )) )) =0;

END_RULE; -- part_requires_project_order
(*)
part_to_approval
*)
RULE part_to_approval FOR (product_definition_formation,
    applied_approval_assignment);

WHERE
    wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
        SIZEOF(QUERY ( fbppa <* applied_approval_assignment | (pdf IN
        fbppa.items) )) <= 1)) )) = 0;

END_RULE; -- part_to_approval
(*)
part_requires_security_classification
*)
RULE part_requires_security_classification FOR (
    product_definition,
    applied_security_classification_assignment);

WHERE
    wr1: SIZEOF(QUERY ( pd <* product_definition |
        ((pd.frame_of_reference.name='part definition') AND
        (NOT (SIZEOF(QUERY ( fbppsca <*
            applied_security_classification_assignment |
            (pd.formation IN fbppsca.items) )) =1 )) )) )=0;

END_RULE; -- part_requires_security_classification
(*)
product_requires_version
*)
RULE product_requires_version FOR (product, product_definition_formation);

WHERE
    wr1: SIZEOF(QUERY ( prod <* product | (NOT (SIZEOF(QUERY ( pdf <*
        product_definition_formation | (prod ::= pdf.of_product) )) >=
        1)) )) = 0;

END_RULE; -- product_requires_version
(*)
project_order_requires_approval
*)
RULE project_order_requires_approval FOR (directed_action,
    applied_approval_assignment);

WHERE
```

```

wr1: SIZEOF(QUERY ( po <* QUERY ( da <* directed_action | (da.name =
    'project order') ) | (NOT (SIZEOF(QUERY ( fbppapp <*
    applied_approval_assignment | (po IN fbppapp.items) )) = 1)) ) )
    = 0;

```

```

END_RULE; -- project_order_requires_approval

```

```

(*)

```

## **project\_order\_tracking\_relationships**

```

(*)

```

```

RULE project_order_tracking_relationships FOR (directed_action,
    action_relationship);

```

```

WHERE

```

```

wr1: SIZEOF(QUERY ( da <* directed_action | ((da.name IN [
    'shop work order','resource acquisition order',
    'digital technical data package work order',
    'pedigree creation order']) AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.related_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    relating_action)) AND (ar.relatng_action.name =
    'project order')))) ) = 1))) ) = 0;
wr2: SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'project order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.relatng_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    related_action)) AND (ar.related_action.name =
    'shop work order')))) ) <= 1))) ) = 0;
wr3: SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'project order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.relatng_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    related_action)) AND (ar.related_action.name =
    'resource acquisition order')))) ) <= 1))) ) = 0;
wr4: SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'project order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.relatng_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    related_action)) AND (ar.related_action.name =
    'digital technical data package work order')))) ) <= 1))) ) =
    0;
wr5: SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'project order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.relatng_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    related_action)) AND (ar.related_action.name =
    'pedigree creation order')))) ) <= 1))) ) = 0;
wr6: SIZEOF(QUERY ( da <* directed_action | ((da.name =
    'customer order') AND (NOT (SIZEOF(QUERY ( ar <*
    action_relationship | ((da :=: ar.related_action) AND ((
    'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
    relating_action)) AND (ar.relatng_action.name =
    'project order')))) ) >= 1))) ) = 0;
wr7: SIZEOF(QUERY ( da <* directed_action | ((da.name =

```

```
'project order') AND (NOT (SIZEOF(QUERY ( ar <*
action_relationship | ((da :=: ar.relateing_action) AND ((
'CAST_PARTS_SCHEMA.DIRECTED_ACTION' IN TYPEOF(ar.
related_action)) AND (ar.related_action.name =
'customer order')))) <= 1))) = 0;
```

```
END_RULE; -- project_order_tracking_relationships
(*)
```

### **restrict\_approval\_status**

\*)

```
RULE restrict_approval_status FOR (approval_status);
```

WHERE

```
wr1: SIZEOF(QUERY ( ast <* approval_status | (NOT (ast.name IN [
'approved','not yet approved','disapproved','withdrawn']))))
= 0;
```

```
END_RULE; -- restrict_approval_status
(*)
```

### **restrict\_externally\_defined\_feature\_definition**

\*)

```
RULE restrict_externally_defined_feature_definition FOR (
externally_defined_feature_definition);
```

WHERE

```
wr1: (SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
(NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS') |
('CAST_PARTS_SCHEMA.DOCUMENT_FILE'
IN TYPEOF(adr.assigned_document)) >= 1)) = 0) OR
(SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
(NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
'CAST_PARTS_SCHEMA.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS') |
('CAST_PARTS_SCHEMA.DOCUMENT_FILE' IN TYPEOF
(adr\document_usage_constraint_assignment.assigned_document_usage.source)
)) >= 1)) = 0);
```

```
END_RULE; -- restrict_externally_defined_feature_definition
(*)
```

### **restrict\_name\_for\_known\_source**

\*)

```
RULE restrict_name_for_known_source FOR (known_source);
```

WHERE

```
wr1: SIZEOF(QUERY ( ks <* known_source | ((ks.name :<>:
'ISO 13584 library') AND (ks.name :<>: 'ISO 10303 part')))) =
0;
```

```
END_RULE; -- restrict_name_for_known_source
(*)
```

### **restrict\_security\_classification\_level**

\*)

```

RULE restrict_security_classification_level FOR (
    security_classification_level);

WHERE
    wr1: SIZEOF(QUERY ( scl <* security_classification_level | (NOT (scl.
        name IN ['unclassified','classified','proprietary',
            'confidential','secret','top_secret']))) ) = 0;

END_RULE; -- restrict_security_classification_level
(*)
subtype_mandatory_characterized_object
*)
RULE subtype_mandatory_characterized_object FOR (characterized_object);

WHERE
    wr1: SIZEOF(QUERY ( csa <* characterized_object | ((NOT (SIZEOF([
        'CAST_PARTS_SCHEMA.DOCUMENT_FILE',
        'CAST_PARTS_SCHEMA.FEATURE_DEFINITION',
        'CAST_PARTS_SCHEMA.CASTING_FEATURE_DEFINITION',
        'CAST_PARTS_SCHEMA.FEATURE_COMPONENT_DEFINITION',
        'CAST_PARTS_SCHEMA.ORDERED_PART'] * TYPEOF(csa)) = 1)) AND (
        NOT (SIZEOF(QUERY ( pd <* USEDIN(csa,
            'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
            'CAST_PARTS_SCHEMA.MATERIAL_PROPERTY' IN TYPEOF(pd)) ) = 1)))
    ))
    = 0;

END_RULE; -- subtype_mandatory_characterized_object
(*)
transition_feature_life_cycle
*)
RULE transition_feature_life_cycle FOR (transition_feature);

WHERE
    wr1: SIZEOF(QUERY ( tf <* transition_feature | (NOT (tf.of_shape.
        definition.frame_of_reference.life_cycle_stage =
            'manufacturing_planning')) ) = 0;

END_RULE; -- transition_feature_life_cycle
(*)
transition_feature_on_part_boundary
*)
RULE transition_feature_on_part_boundary FOR (transition_feature);

WHERE
    wr1: SIZEOF(QUERY ( tf <* transition_feature | (NOT tf.
        product_definitional) ) = 0;

END_RULE; -- transition_feature_on_part_boundary
(*)
acyclic_mapped_representation
*)

```

```

FUNCTION acyclic_mapped_representation(
    parent_set: SET OF representation;
    children_set: SET OF representation_item
): BOOLEAN;

LOCAL
    x : SET OF representation_item;
    y : SET OF representation_item;
END_LOCAL;
x := QUERY ( z <* children_set | ('CAST_PARTS_SCHEMA.MAPPED_ITEM' IN
    TYPEOF(z)) );
IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
        IF x[i]\mapped_item.mapping_source.mapped_representation IN
            parent_set THEN
            RETURN(FALSE);
        END_IF;
        IF NOT acyclic_mapped_representation(parent_set + x[i]\mapped_item
            .mapping_source.mapped_representation,x[i]\mapped_item.
            mapping_source.mapped_representation.items) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_IF;
x := children_set - x;
IF SIZEOF(x) > 0 THEN
    REPEAT i := 1 TO HIINDEX(x) BY 1;
        y := QUERY ( z <* bag_to_set(USEDIN(x[i], '')) | (
            'CAST_PARTS_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) );
        IF NOT acyclic_mapped_representation(parent_set,y) THEN
            RETURN(FALSE);
        END_IF;
    END_REPEAT;
END_IF;
RETURN(TRUE);

END_FUNCTION; -- acyclic_mapped_representation

```

(\*)

### **acyclic\_product\_definition\_relationship**

\*)

```

FUNCTION acyclic_product_definition_relationship(
    relation: product_definition_relationship;
    relatives: SET [1:?] OF product_definition;
    specific_relation: STRING
): BOOLEAN;

LOCAL
    x : SET OF product_definition_relationship;
END_LOCAL;
IF relation.relatng_product_definition IN relatives THEN
    RETURN(FALSE);
END_IF;
x := QUERY ( pd <* bag_to_set(USEDIN(relation.

```

```

    relating_product_definition, ('CAST_PARTS_SCHEMA.' +
    'PRODUCT_DEFINITION_RELATIONSHIP.') + 'RELATED_PRODUCT_DEFINITION'))
    | (specific_relation IN TYPEOF(pd)) );
REPEAT i := 1 TO HIINDEX(x) BY 1;
    IF NOT acyclic_product_definition_relationship(x[i], relatives +
        relation.relativing_product_definition, specific_relation) THEN
        RETURN(FALSE);
    END_IF;
END_REPEAT;
RETURN(TRUE);

```

```
END_FUNCTION; -- acyclic_product_definition_relationship
```

(\*

### **associated\_surface**

\*)

```

FUNCTION associated_surface(
    arg: pcurve_or_surface
): surface;

LOCAL
    surf : surface;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(arg) THEN
    surf := arg.basis_surface;
ELSE
    surf := arg;
END_IF;
RETURN(surf);

```

```
END_FUNCTION; -- associated_surface
```

(\*

### **bag\_to\_set**

\*)

```

FUNCTION bag_to_set(
    the_bag: BAG OF GENERIC:intype
): SET OF GENERIC:intype;

LOCAL
    the_set : SET OF GENERIC:intype := [];
END_LOCAL;
IF SIZEOF(the_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(the_bag) BY 1;
        the_set := the_set + the_bag[i];
    END_REPEAT;
END_IF;
RETURN(the_set);

```

```
END_FUNCTION; -- bag_to_set
```

(\*

### **base\_axis**

\*)

```
FUNCTION base_axis(
```

```

        dim: INTEGER;
        axis1, axis2, axis3: direction
    ): LIST [2:3] OF direction;

LOCAL
    u      : LIST [2:3] OF direction;
    d1     : direction;
    d2     : direction;
    factor : REAL;
END_LOCAL;
IF dim = 3 THEN
    d1 := NVL(normalise(axis3), dummy_gri || direction([0,0,1]));
    d2 := first_proj_axis(d1, axis1);
    u := [d2, second_proj_axis(d1, d2, axis2), d1];
ELSE
    IF EXISTS(axis1) THEN
        d1 := normalise(axis1);
        u := [d1, orthogonal_complement(d1)];
        IF EXISTS(axis2) THEN
            factor := dot_product(axis2, u[2]);
            IF factor < 0 THEN
                u[2].direction_ratios[1] := -u[2].direction_ratios[1];
                u[2].direction_ratios[2] := -u[2].direction_ratios[2];
            END_IF;
        END_IF;
    ELSE
        IF EXISTS(axis2) THEN
            d1 := normalise(axis2);
            u := [orthogonal_complement(d1), d1];
            u[1].direction_ratios[1] := -u[1].direction_ratios[1];
            u[1].direction_ratios[2] := -u[1].direction_ratios[2];
        ELSE
            u := [dummy_gri || direction([1,0]), dummy_gri || direction([0,1])];
        END_IF;
    END_IF;
END_IF;
RETURN(u);

END_FUNCTION; -- base_axis
(*)
boolean_choose
*)
FUNCTION boolean_choose(
    b: BOOLEAN;
    choicel, choice2: GENERIC:item
): GENERIC:item;
IF b THEN
    RETURN(choicel);
ELSE
    RETURN(choice2);
END_IF;

END_FUNCTION; -- boolean_choose

```



(\*)

**build\_2axes**

\*)

```

FUNCTION build_2axes(
    ref_direction: direction
): LIST [2:2] OF direction;

LOCAL
    d : direction := NVL(normalise(ref_direction), dummy_gri ||
        direction([1,0]));
END_LOCAL;
RETURN([d, orthogonal_complement(d)]);

END_FUNCTION; -- build_2axes

```

(\*)

**build\_axes**

\*)

```

FUNCTION build_axes(
    axis, ref_direction: direction
): LIST [3:3] OF direction;

LOCAL
    d1 : direction;
    d2 : direction;
END_LOCAL;
d1 := NVL(normalise(axis), dummy_gri || direction([0,0,1]));
d2 := first_proj_axis(d1, ref_direction);
RETURN([d2, normalise(cross_product(d1,d2)).orientation, d1]);

END_FUNCTION; -- build_axes

```

(\*)

**closed\_shell\_reversed**

\*)

```

FUNCTION closed_shell_reversed(
    a_shell: closed_shell
): oriented_closed_shell;

LOCAL
    the_reverse : oriented_closed_shell;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.ORIENTED_CLOSED_SHELL' IN TYPEOF(a_shell) THEN
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces)) || closed_shell()) ||
        oriented_closed_shell(a_shell\oriented_closed_shell.
            closed_shell_element, NOT a_shell\oriented_closed_shell.
                orientation);
ELSE
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces)) || closed_shell()) ||
        oriented_closed_shell(a_shell, FALSE);
END_IF;
RETURN(the_reverse);

```

```

END_FUNCTION; -- closed_shell_reversed
(*)
conditional_reverse
*)
FUNCTION conditional_reverse(
    p: BOOLEAN;
    an_item: reversible_topology
): reversible_topology;
IF p THEN
    RETURN(an_item);
ELSE
    RETURN(topology_reversed(an_item));
END_IF;

END_FUNCTION; -- conditional_reverse
(*)
constraints_composite_curve_on_surface
*)
FUNCTION constraints_composite_curve_on_surface(
    c: composite_curve_on_surface
): BOOLEAN;

LOCAL
    n_segments : INTEGER := SIZEOF(c.segments);
END_LOCAL;
REPEAT k := 1 TO n_segments BY 1;
    IF ((NOT ('CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(c.composite_curve.
        segments[k].parent_curve))) AND (NOT (
        'CAST_PARTS_SCHEMA.SURFACE_CURVE' IN TYPEOF(c.composite_curve.
        segments[k].parent_curve)))) AND (NOT (
        'CAST_PARTS_SCHEMA.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) THEN
        RETURN(FALSE);
    END_IF;
END_REPEAT;
RETURN(TRUE);

END_FUNCTION; -- constraints_composite_curve_on_surface
(*)
constraints_param_b_spline
*)
FUNCTION constraints_param_b_spline(
    degree, up_knots, up_cp: INTEGER;
    knot_mult: LIST OF INTEGER;
    knots: LIST OF parameter_value
): BOOLEAN;

LOCAL
    k      : INTEGER;
    sum    : INTEGER;
    result : BOOLEAN := TRUE;

```

```

END_LOCAL;
sum := knot_mult[1];
REPEAT i := 2 TO up_knots BY 1;
  sum := sum + knot_mult[i];
END_REPEAT;
IF (((degree < 1) OR (up_knots < 2)) OR (up_cp < degree)) OR (sum <> (
  (degree + up_cp) + 2)) THEN
  result := FALSE;
  RETURN(result);
END_IF;
k := knot_mult[1];
IF (k < 1) OR (k > (degree + 1)) THEN
  result := FALSE;
  RETURN(result);
END_IF;
REPEAT i := 2 TO up_knots BY 1;
  IF (knot_mult[i] < 1) OR (knots[i] <= knots[i - 1]) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  k := knot_mult[i];
  IF (i < up_knots) AND (k > degree) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
  IF (i = up_knots) AND (k > (degree + 1)) THEN
    result := FALSE;
    RETURN(result);
  END_IF;
END_REPEAT;
RETURN(result);

```

```
END_FUNCTION; -- constraints_param_b_spline
```

```
(*
```

## **cross\_product**

```
*)
```

```

FUNCTION cross_product(
  arg1, arg2: direction
): vector;

LOCAL
  v2 : LIST [3:3] OF REAL;
  v1 : LIST [3:3] OF REAL;
  mag : REAL;
  res : direction;
  result : vector;
END_LOCAL;
IF ((NOT EXISTS(arg1)) OR (arg1.dim = 2)) OR ((NOT EXISTS(arg2)) OR (
  arg2.dim = 2)) THEN
  RETURN(?);
ELSE
  BEGIN
    v1 := normalise(arg1).direction_ratios;

```

```

v2 := normalise(arg2).direction_ratios;
res := dummy_gri || direction([(v1[2] * v2[3]) - (v1[3] * v2[2]), (
    v1[3] * v2[1]) - (v1[1] * v2[3]), (v1[1] * v2[2]) - (v1[2] * v2[
    1])]);
mag := 0;
REPEAT i := 1 TO 3 BY 1;
    mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
END_REPEAT;
IF mag > 0 THEN
    result := dummy_gri || vector(res, SQRT(mag));
ELSE
    result := dummy_gri || vector(arg1, 0);
END_IF;
RETURN(result);
END;
END_IF;

```

```

END_FUNCTION; -- cross_product
(*)

```

### **curve\_weights\_positive**

```

*)
FUNCTION curve_weights_positive(
    b: rational_b_spline_curve
): BOOLEAN;

LOCAL
    result : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 0 TO b.upper_index_on_control_points BY 1;
    IF b.weights[i] <= 0 THEN
        result := FALSE;
        RETURN(result);
    END_IF;
END_REPEAT;
RETURN(result);

```

```

END_FUNCTION; -- curve_weights_positive
(*)

```

### **derive\_dimensional\_exponents**

```

*)
FUNCTION derive_dimensional_exponents(
    x: unit
): dimensional_exponents;

LOCAL
    result : dimensional_exponents := dimensional_exponents(0, 0, 0, 0, 0, 0,
    0);
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.DERIVED_UNIT' IN TYPEOF(x) THEN
    REPEAT i := LOINDEX(x.elements) TO HIINDEX(x.elements) BY 1;
        result.length_exponent := result.length_exponent + (x.elements[i].
        exponent * x.elements[i].unit.dimensions.length_exponent);
        result.mass_exponent := result.mass_exponent + (x.elements[i].

```

```

        exponent * x.elements[i].unit.dimensions.mass_exponent);
result.time_exponent := result.time_exponent + (x.elements[i].
    exponent * x.elements[i].unit.dimensions.time_exponent);
result.electric_current_exponent := result.
    electric_current_exponent + (x.elements[i].exponent * x.
    elements[i].unit.dimensions.electric_current_exponent);
result.thermodynamic_temperature_exponent := result.
    thermodynamic_temperature_exponent + (x.elements[i].exponent *
    x.elements[i].unit.dimensions.
    thermodynamic_temperature_exponent);
result.amount_of_substance_exponent := result.
    amount_of_substance_exponent + (x.elements[i].exponent * x.
    elements[i].unit.dimensions.amount_of_substance_exponent);
result.luminous_intensity_exponent := result.
    luminous_intensity_exponent + (x.elements[i].exponent * x.
    elements[i].unit.dimensions.luminous_intensity_exponent);
    END_REPEAT;
ELSE
    result := x.dimensions;
END_IF;
RETURN(result);

END_FUNCTION; -- derive_dimensional_exponents
(*)
dimension_of
*)
FUNCTION dimension_of(
    item: geometric_representation_item
): dimension_count;

LOCAL
    x : SET OF representation;
    y : representation_context;
    dim : dimension_count;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.CARTESIAN_POINT' IN TYPEOF(item) THEN
    dim := SIZEOF(item\cartesian_point.coordinates);
    RETURN(dim);
END_IF;
IF 'CAST_PARTS_SCHEMA.DIRECTION' IN TYPEOF(item) THEN
    dim := SIZEOF(item\direction.direction_ratios);
    RETURN(dim);
END_IF;
IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(item) THEN
    dim := SIZEOF(item\vector.orientation\direction.direction_ratios);
    RETURN(dim);
END_IF;
x := using_representations(item);
y := x[1].context_of_items;
dim := y\geometric_representation_context.coordinate_space_dimension;
RETURN(dim);

END_FUNCTION; -- dimension_of

```

(\*)

**dimensions\_for\_si\_unit**

\*)

```

FUNCTION dimensions_for_si_unit(
    n: si_unit_name
): dimensional_exponents;
CASE n OF
    metre      : RETURN(dimensional_exponents(1,0,0,0,0,0,0));
    gram       : RETURN(dimensional_exponents(0,1,0,0,0,0,0));
    second     : RETURN(dimensional_exponents(0,0,1,0,0,0,0));
    ampere     : RETURN(dimensional_exponents(0,0,0,1,0,0,0));
    kelvin     : RETURN(dimensional_exponents(0,0,0,0,1,0,0));
    mole       : RETURN(dimensional_exponents(0,0,0,0,0,1,0));
    candela    : RETURN(dimensional_exponents(0,0,0,0,0,0,1));
    radian     : RETURN(dimensional_exponents(0,0,0,0,0,0,0));
    steradian  : RETURN(dimensional_exponents(0,0,0,0,0,0,0));
    hertz      : RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
    newton     : RETURN(dimensional_exponents(1,1,-2,0,0,0,0));
    pascal     : RETURN(dimensional_exponents(-1,1,-2,0,0,0,0));
    joule      : RETURN(dimensional_exponents(2,1,-2,0,0,0,0));
    watt       : RETURN(dimensional_exponents(2,1,-3,0,0,0,0));
    coulomb    : RETURN(dimensional_exponents(0,0,1,1,0,0,0));
    volt       : RETURN(dimensional_exponents(2,1,-3,-1,0,0,0));
    farad      : RETURN(dimensional_exponents(-2,-1,4,1,0,0,0));
    ohm        : RETURN(dimensional_exponents(2,1,-3,-2,0,0,0));
    siemens    : RETURN(dimensional_exponents(-2,-1,3,2,0,0,0));
    weber      : RETURN(dimensional_exponents(2,1,-2,-1,0,0,0));
    tesla      : RETURN(dimensional_exponents(0,1,-2,-1,0,0,0));
    henry      : RETURN(dimensional_exponents(2,1,-2,-2,0,0,0));
    degree_celsius : RETURN(dimensional_exponents(0,0,0,0,1,0,0));
    lumen      : RETURN(dimensional_exponents(0,0,0,0,0,0,1));
    lux        : RETURN(dimensional_exponents(-2,0,0,0,0,0,1));
    becquerel  : RETURN(dimensional_exponents(0,0,-1,0,0,0,0));
    gray       : RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
    sievert    : RETURN(dimensional_exponents(2,0,-2,0,0,0,0));
END_CASE;

END_FUNCTION; -- dimensions_for_si_unit

```

(\*)

**dot\_product**

\*)

```

FUNCTION dot_product(
    arg1, arg2: direction
): REAL;

LOCAL
    ndim  : INTEGER;
    scalar : REAL;
    vec1  : direction;
    vec2  : direction;
END_LOCAL;
IF (NOT EXISTS(arg1)) OR (NOT EXISTS(arg2)) THEN
    scalar := ?;

```

```

ELSE
  IF arg1.dim <> arg2.dim THEN
    scalar := ?;
  ELSE
    BEGIN
      vec1 := normalise(arg1);
      vec2 := normalise(arg2);
      ndim := arg1.dim;
      scalar := 0;
      REPEAT i := 1 TO ndim BY 1;
        scalar := scalar + (vec1.direction_ratios[i] * vec2.
          direction_ratios[i]);
      END_REPEAT;
    END;
  END_IF;
END_IF;
RETURN(scalar);

```

```
END_FUNCTION; -- dot_product
```

(\*

### **edge\_reversed**

\*)

```

FUNCTION edge_reversed(
  an_edge: edge
): oriented_edge;

LOCAL
  the_reverse : oriented_edge;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.ORIENTED_EDGE' IN TYPEOF(an_edge) THEN
  the_reverse := (dummy_tri || edge(an_edge.edge_end,an_edge.
    edge_start)) || oriented_edge(an_edge\oriented_edge.edge_element,
    NOT an_edge\oriented_edge.orientation);
ELSE
  the_reverse := (dummy_tri || edge(an_edge.edge_end,an_edge.
    edge_start)) || oriented_edge(an_edge,FALSE);
END_IF;
RETURN(the_reverse);

```

```
END_FUNCTION; -- edge_reversed
```

(\*

### **face\_bound\_reversed**

\*)

```

FUNCTION face_bound_reversed(
  a_face_bound: face_bound
): face_bound;

LOCAL
  the_reverse : face_bound;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(a_face_bound) THEN
  the_reverse := (dummy_tri || face_bound(a_face_bound\face_bound.
    bound,NOT a_face_bound\face_bound.orientation)) ||

```

```

        face_outer_bound();
    ELSE
        the_reverse := dummy_tri || face_bound(a_face_bound.bound,NOT
        a_face_bound.orientation);
    END_IF;
    RETURN(the_reverse);

END_FUNCTION; -- face_bound_reversed
(*)
face_reversed
*)
FUNCTION face_reversed(
    a_face: face
): oriented_face;

LOCAL
    the_reverse : oriented_face;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.ORIENTED_FACE' IN TYPEOF(a_face) THEN
    the_reverse := (dummy_tri || face(set_of_topology_reversed(a_face.
    bounds)) || oriented_face(a_face\oriented_face.face_element,NOT
    a_face\oriented_face.orientation));
ELSE
    the_reverse := (dummy_tri || face(set_of_topology_reversed(a_face.
    bounds)) || oriented_face(a_face,FALSE));
END_IF;
RETURN(the_reverse);

END_FUNCTION; -- face_reversed
(*)
first_proj_axis
*)
FUNCTION first_proj_axis(
    z_axis, arg: direction
): direction;

LOCAL
    x_vec : vector;
    v      : direction;
    z      : direction;
    x_axis : direction;
END_LOCAL;
IF NOT EXISTS(z_axis) THEN
    RETURN(?);
ELSE
    z := normalise(z_axis);
    IF NOT EXISTS(arg) THEN
        IF z.direction_ratios <> [1,0,0] THEN
            v := dummy_gri || direction([1,0,0]);
        ELSE
            v := dummy_gri || direction([0,1,0]);
        END_IF;
    ELSE
        v := arg;
    END_IF;
    RETURN(v);
END_IF;
END_FUNCTION;

```



```

    IF arg.dim <> 3 THEN
        RETURN(?);
    END_IF;
    IF cross_product(arg,z).magnitude = 0 THEN
        RETURN(?);
    ELSE
        v := normalise(arg);
    END_IF;
END_IF;
x_vec := scalar_times_vector(dot_product(v,z),z);
x_axis := vector_difference(v,x_vec).orientation;
x_axis := normalise(x_axis);
END_IF;
RETURN(x_axis);

END_FUNCTION; -- first_proj_axis
(*)
get_basis_surface
*)
FUNCTION get_basis_surface(
    c: curve_on_surface
): SET [0:2] OF surface;

LOCAL
    surfs : SET [0:2] OF surface;
    n      : INTEGER;
END_LOCAL;
surfs := [];
IF 'CAST_PARTS_SCHEMA.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
ELSE
    IF 'CAST_PARTS_SCHEMA.SURFACE_CURVE' IN TYPEOF(c) THEN
        n := SIZEOF(c\surface_curve.associated_geometry);
        REPEAT i := 1 TO n BY 1;
            surfs := surfs + associated_surface(c\surface_curve.
                associated_geometry[i]);
        END_REPEAT;
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.COMPOSITE_CURVE_ON_SURFACE' IN TYPEOF(c) THEN
    n := SIZEOF(c\composite_curve.segments);
    surfs := get_basis_surface(c\composite_curve.segments[1].
        parent_curve);
    IF n > 1 THEN
        REPEAT i := 2 TO n BY 1;
            surfs := surfs * get_basis_surface(c\composite_curve.segments[i]
                .parent_curve);
        END_REPEAT;
    END_IF;
END_IF;
RETURN(surfs);

END_FUNCTION; -- get_basis_surface

```

(\*

**get\_description\_value**

\*)

```

FUNCTION get_description_value(
    obj: description_attribute_select
): text;

LOCAL
    description_bag : BAG OF description_attribute := USEDIN(obj, (
        'CAST_PARTS_SCHEMA.' + 'DESCRIPTION_ATTRIBUTE.') +
        'DESCRIBED_ITEM');
END_LOCAL;
IF SIZEOF(description_bag) = 1 THEN
    RETURN(description_bag[1].attribute_value);
ELSE
    RETURN(?);
END_IF;

END_FUNCTION; -- get_description_value

```

(\*

**get\_id\_value**

\*)

```

FUNCTION get_id_value(
    obj: id_attribute_select
): identifier;

LOCAL
    id_bag : BAG OF id_attribute := USEDIN(obj, ('CAST_PARTS_SCHEMA.' +
        'ID_ATTRIBUTE.') + 'IDENTIFIED_ITEM');
END_LOCAL;
IF SIZEOF(id_bag) = 1 THEN
    RETURN(id_bag[1].attribute_value);
ELSE
    RETURN(?);
END_IF;

END_FUNCTION; -- get_id_value

```

(\*

**get\_name\_value**

\*)

```

FUNCTION get_name_value(
    obj: name_attribute_select
): label;

LOCAL
    name_bag : BAG OF name_attribute := USEDIN(obj, ('CAST_PARTS_SCHEMA.'
        + 'NAME_ATTRIBUTE.') + 'NAMED_ITEM');
END_LOCAL;
IF SIZEOF(name_bag) = 1 THEN
    RETURN(name_bag[1].attribute_value);
ELSE
    RETURN(?);
END_IF;

```

```

    END_IF;

    END_FUNCTION; -- get_name_value
  (*
get_property_definition_representations
  *)
  FUNCTION get_property_definition_representations(
      c_def_instance: characterized_definition
  ): SET OF property_definition_representation;

  LOCAL
    pdr_set : SET OF property_definition_representation := [];
    pd_set  : SET OF property_definition := [];
  END_LOCAL;
  pd_set := bag_to_set(USEDIN(c_def_instance,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION'));
  IF SIZEOF(pd_set) < 1 THEN
    RETURN(pdr_set);
  END_IF;
  REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
    pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'));
  END_REPEAT;
  RETURN(pdr_set);

  END_FUNCTION; -- get_property_definition_representations
  (*
get_role
  *)
  FUNCTION get_role(
      obj: role_select
  ): object_role;

  LOCAL
    role_bag : BAG OF role_association := USEDIN(obj, (
    'CAST_PARTS_SCHEMA.' + 'ROLE_ASSOCIATION.') +
    'ITEM_WITH_ROLE');
  END_LOCAL;
  IF SIZEOF(role_bag) = 1 THEN
    RETURN(role_bag[1].role);
  ELSE
    RETURN(?);
  END_IF;

  END_FUNCTION; -- get_role
  (*
item_in_context
  *)
  FUNCTION item_in_context(
      item: representation_item;
      cntxt: representation_context
  ): BOOLEAN;

```

```

LOCAL
  y : BAG OF representation_item;
END_LOCAL;
IF SIZEOF(USEDIN(item, 'CAST_PARTS_SCHEMA.REPRESENTATION.ITEMS') *
  cntxt.representations_in_context) > 0 THEN
  RETURN(TRUE);
ELSE
  y := QUERY ( z <* USEDIN(item, '') | (
    'CAST_PARTS_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) );
  IF SIZEOF(y) > 0 THEN
    REPEAT i := 1 TO HIINDEX(y) BY 1;
      IF item_in_context(y[i], cntxt) THEN
        RETURN(TRUE);
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(FALSE);
END_FUNCTION; -- item_in_context
(*)
leap_year
*)
FUNCTION leap_year(
  year: year_number
): BOOLEAN;
IF ((year MOD 4) = 0) AND ((year MOD 100) <> 0) OR ((year MOD 400) =
  0) THEN
  RETURN(TRUE);
ELSE
  RETURN(FALSE);
END_IF;
END_FUNCTION; -- leap_year
(*)
list_face_loops
*)
FUNCTION list_face_loops(
  f: face
): LIST [0:?] OF loop;

LOCAL
  loops : LIST [0:?] OF loop := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(f.bounds) BY 1;
  loops := loops + f.bounds[i].bound;
END_REPEAT;
RETURN(loops);

END_FUNCTION; -- list_face_loops
(*)

```

**list\_of\_topology\_reversed**

```

*)
FUNCTION list_of_topology_reversed(
    a_list: list_of_reversible_topology_item
): list_of_reversible_topology_item;

LOCAL
    the_reverse : list_of_reversible_topology_item;
END_LOCAL;
the_reverse := [];
REPEAT i := 1 TO SIZEOF(a_list) BY 1;
    the_reverse := topology_reversed(a_list[i]) + the_reverse;
END_REPEAT;
RETURN(the_reverse);

END_FUNCTION; -- list_of_topology_reversed

```

(\*)

**list\_to\_array**

```

*)
FUNCTION list_to_array(
    lis: LIST [0:?] OF GENERIC:t;
    low, u: INTEGER
): ARRAY OF GENERIC:t;

LOCAL
    n : INTEGER;
    res : ARRAY [low:u] OF GENERIC:t;
END_LOCAL;
n := SIZEOF(lis);
IF n <> ((u - low) + 1) THEN
    RETURN(?);
ELSE
    res := [lis[1],n];
    REPEAT i := 2 TO n BY 1;
        res[(low + i) - 1] := lis[i];
    END_REPEAT;
    RETURN(res);
END_IF;

END_FUNCTION; -- list_to_array

```

(\*)

**list\_to\_set**

```

*)
FUNCTION list_to_set(
    l: LIST [0:?] OF GENERIC:t
): SET OF GENERIC:t;

LOCAL
    s : SET OF GENERIC:t := [];
END_LOCAL;
REPEAT i := 1 TO SIZEOF(l) BY 1;
    s := s + l[i];

```

```
END_REPEAT;
RETURN(s);
```

```
END_FUNCTION; -- list_to_set
(*)
```

### **make\_array\_of\_array**

```
*)
```

```
FUNCTION make_array_of_array(
    lis: LIST [1:?] OF LIST [1:?] OF GENERIC:t;
    low1, u1, low2, u2: INTEGER
): ARRAY OF ARRAY OF GENERIC:t;

LOCAL
    res : ARRAY [low1:u1] OF ARRAY [low2:u2] OF GENERIC:t;
END_LOCAL;
IF ((u1 - low1) + 1) <> SIZEOF(lis) THEN
    RETURN(?);
END_IF;
IF ((u2 - low2) + 1) <> SIZEOF(lis[1]) THEN
    RETURN(?);
END_IF;
res := [list_to_array(lis[1], low2, u2), (u1 - low1) + 1];
REPEAT i := 2 TO HIINDEX(lis) BY 1;
    IF ((u2 - low2) + 1) <> SIZEOF(lis[i]) THEN
        RETURN(?);
    END_IF;
    res[(low1 + i) - 1] := list_to_array(lis[i], low2, u2);
END_REPEAT;
RETURN(res);
```

```
END_FUNCTION; -- make_array_of_array
(*)
```

### **mixed\_loop\_type\_set**

```
*)
```

```
FUNCTION mixed_loop_type_set(
    l: SET [0:?] OF loop
): LOGICAL;

LOCAL
    poly_loop_type : LOGICAL;
END_LOCAL;
IF SIZEOF(l) <= 1 THEN
    RETURN(FALSE);
END_IF;
poly_loop_type := 'CAST_PARTS_SCHEMA.POLY_LOOP' IN TYPEOF(l[1]);
REPEAT i := 2 TO SIZEOF(l) BY 1;
    IF ('CAST_PARTS_SCHEMA.POLY_LOOP' IN TYPEOF(l[i])) <> poly_loop_type
        THEN
            RETURN(TRUE);
        END_IF;
END_REPEAT;
RETURN(FALSE);
END_FUNCTION; -- mixed_loop_type_set
```

(\*

## **msb\_shells**

\*)

```

FUNCTION msb_shells(
    brep: manifold_solid_brep
): SET [1:?] OF closed_shell;
IF SIZEOF(QUERY ( msbtype <* TYPEOF(brep) | (msbtype LIKE
    '*BREP_WITH_VOID') )) >= 1 THEN
    RETURN(brep\brep_with_voids.voids + brep.outer);
ELSE
    RETURN([brep.outer]);
END_IF;

END_FUNCTION; -- msb_shells

```

(\*

## **normalise**

\*)

```

FUNCTION normalise(
    arg: vector_or_direction
): vector_or_direction;

LOCAL
    ndim    : INTEGER;
    v       : direction;
    vec     : vector;
    mag     : REAL;
    result  : vector_or_direction;
END_LOCAL;
IF NOT EXISTS(arg) THEN
    result := ?;
ELSE
    ndim := arg.dim;
    IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
        BEGIN
            v := dummy_gri || direction(arg.orientation.direction_ratios);
            IF arg.magnitude = 0 THEN
                RETURN(?);
            ELSE
                vec := dummy_gri || vector(v,1);
                END_IF;
            END;
        ELSE
            v := dummy_gri || direction(arg.direction_ratios);
            END_IF;
            mag := 0;
            REPEAT i := 1 TO ndim BY 1;
                mag := mag + (v.direction_ratios[i] * v.direction_ratios[i]);
            END_REPEAT;
            IF mag > 0 THEN
                mag := SQRT(mag);
                REPEAT i := 1 TO ndim BY 1;
                    v.direction_ratios[i] := v.direction_ratios[i] / mag;
                END_REPEAT;
            END_IF;
        END;
    END;
END;

```

```

        IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(arg) THEN
            vec.orientation := v;
            result := vec;
        ELSE
            result := v;
        END_IF;
    ELSE
        RETURN(?);
    END_IF;
END_IF;
RETURN(result);

```

```
END_FUNCTION; -- normalise
```

(\*

### **open\_shell\_reversed**

\*)

```

FUNCTION open_shell_reversed(
    a_shell: open_shell
): oriented_open_shell;

LOCAL
    the_reverse : oriented_open_shell;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.ORIENTED_OPEN_SHELL' IN TYPEOF(a_shell) THEN
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces)) || open_shell()) ||
        oriented_open_shell(a_shell\oriented_open_shell.
            open_shell_element,NOT a_shell\oriented_open_shell.orientation);
ELSE
    the_reverse := ((dummy_tri || connected_face_set(a_shell\
        connected_face_set.cfs_faces)) || open_shell()) ||
        oriented_open_shell(a_shell,FALSE);
END_IF;
RETURN(the_reverse);

```

```
END_FUNCTION; -- open_shell_reversed
```

(\*

### **orthogonal\_complement**

\*)

```

FUNCTION orthogonal_complement(
    vec: direction
): direction;

LOCAL
    result : direction;
END_LOCAL;
IF (vec.dim <> 2) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
ELSE
    result := dummy_gri || direction([-vec.direction_ratios[2],vec.
        direction_ratios[1]]);
    RETURN(result);
END_IF;

```



```

END_FUNCTION; -- orthogonal_complement
(*)
path_head_to_tail
*)
FUNCTION path_head_to_tail(
    a_path: path
): BOOLEAN;

LOCAL
    n : INTEGER;
    p : BOOLEAN := TRUE;
END_LOCAL;
n := SIZEOF(a_path.edge_list);
REPEAT i := 2 TO n BY 1;
    p := p AND (a_path.edge_list[i - 1].edge_end ::= a_path.edge_list[i]
        .edge_start);
END_REPEAT;
RETURN(p);

END_FUNCTION; -- path_head_to_tail

```

```

(*)
path_reverse
*)
FUNCTION path_reversed(
    a_path: path
): oriented_path;

LOCAL
    the_reverse : oriented_path;
END_LOCAL;
IF 'CAST_PARTS_SCHEMA.ORIENTED_PATH' IN TYPEOF(a_path) THEN
    the_reverse := (dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list))) || oriented_path(a_path\oriented_path.path_element,
        NOT a_path\oriented_path.orientation);
ELSE
    the_reverse := (dummy_tri || path(list_of_topology_reversed(a_path.
        edge_list))) || oriented_path(a_path, FALSE);
END_IF;
RETURN(the_reverse);

END_FUNCTION; -- path_reversed

```

```

(*)
scalar_times_vector
*)
FUNCTION scalar_times_vector(
    scalar: REAL;
    vec: vector_or_direction
): vector;

LOCAL
    v : direction;

```

## ISO 10303-223:2008 (E)

```

    mag      : REAL;
    result   : vector;
END_LOCAL;
IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
    RETURN(?);
ELSE
    IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(vec) THEN
        v := dummy_gri || direction(vec.orientation.direction_ratios);
        mag := scalar * vec.magnitude;
    ELSE
        v := dummy_gri || direction(vec.direction_ratios);
        mag := scalar;
    END_IF;
    IF mag < 0 THEN
        REPEAT i := 1 TO SIZEOF(v.direction_ratios) BY 1;
            v.direction_ratios[i] := -v.direction_ratios[i];
        END_REPEAT;
        mag := -mag;
    END_IF;
    result := dummy_gri || vector(normalise(v),mag);
END_IF;
RETURN(result);

END_FUNCTION; -- scalar_times_vector
(*)
second_proj_axis
*)
FUNCTION second_proj_axis(
    z_axis, x_axis, arg: direction
): direction;

LOCAL
    temp    : vector;
    v       : direction;
    y_axis  : vector;
END_LOCAL;
IF NOT EXISTS(arg) THEN
    v := dummy_gri || direction([0,1,0]);
ELSE
    v := arg;
END_IF;
temp := scalar_times_vector(dot_product(v,z_axis),z_axis);
y_axis := vector_difference(v,temp);
temp := scalar_times_vector(dot_product(v,x_axis),x_axis);
y_axis := vector_difference(y_axis,temp);
y_axis := normalise(y_axis);
RETURN(y_axis.orientation);

END_FUNCTION; -- second_proj_axis
(*)
set_of_topology_reversed
*)
FUNCTION set_of_topology_reversed(
1770
```

© ISO 2008—All rights reserved

```

        a_set: set_of_reversible_topology_item
    ): set_of_reversible_topology_item;

LOCAL
    the_reverse : set_of_reversible_topology_item;
END_LOCAL;
the_reverse := [];
REPEAT i := 1 TO SIZEOF(a_set) BY 1;
    the_reverse := the_reverse + topology_reversed(a_set[i]);
END_REPEAT;
RETURN(the_reverse);

END_FUNCTION; -- set_of_topology_reversed
(*)
shell_reversed
*)
FUNCTION shell_reversed(
    a_shell: shell
): shell;
IF 'CAST_PARTS_SCHEMA.OPEN_SHELL' IN TYPEOF(a_shell) THEN
    RETURN(open_shell_reversed(a_shell));
ELSE
    IF 'CAST_PARTS_SCHEMA.CLOSED_SHELL' IN TYPEOF(a_shell) THEN
        RETURN(closed_shell_reversed(a_shell));
    ELSE
        RETURN(?);
    END_IF;
END_IF;

END_FUNCTION; -- shell_reversed
(*)
surface_weights_positive
*)
FUNCTION surface_weights_positive(
    b: rational_b_spline_surface
): BOOLEAN;

LOCAL
    result : BOOLEAN := TRUE;
END_LOCAL;
REPEAT i := 0 TO b.u_upper BY 1;
    REPEAT j := 0 TO b.v_upper BY 1;
        IF b.weights[i][j] <= 0 THEN
            result := FALSE;
            RETURN(result);
        END_IF;
    END_REPEAT;
END_REPEAT;
RETURN(result);

END_FUNCTION; -- surface_weights_positive
(*)

```

**topology\_reversed**

\*)

```

FUNCTION topology_reversed(
    an_item: reversible_topology
): reversible_topology;
IF 'CAST_PARTS_SCHEMA.EDGE' IN TYPEOF(an_item) THEN
    RETURN(edge_reversed(an_item));
END_IF;
IF 'CAST_PARTS_SCHEMA.PATH' IN TYPEOF(an_item) THEN
    RETURN(path_reversed(an_item));
END_IF;
IF 'CAST_PARTS_SCHEMA.FACE_BOUND' IN TYPEOF(an_item) THEN
    RETURN(face_bound_reversed(an_item));
END_IF;
IF 'CAST_PARTS_SCHEMA.FACE' IN TYPEOF(an_item) THEN
    RETURN(face_reversed(an_item));
END_IF;
IF 'CAST_PARTS_SCHEMA.SHELL' IN TYPEOF(an_item) THEN
    RETURN(shell_reversed(an_item));
END_IF;
IF 'SET' IN TYPEOF(an_item) THEN
    RETURN(set_of_topology_reversed(an_item));
END_IF;
IF 'LIST' IN TYPEOF(an_item) THEN
    RETURN(list_of_topology_reversed(an_item));
END_IF;
RETURN(?);

END_FUNCTION; -- topology_reversed

```

(\*)

**using\_items**

\*)

```

FUNCTION using_items(
    item: founded_item_select;
    checked_items: SET OF founded_item_select
): SET OF founded_item_select;

LOCAL
    next_items      : SET OF founded_item_select;
    new_check_items : SET OF founded_item_select;
    result_items    : SET OF founded_item_select;
END_LOCAL;
result_items := [];
new_check_items := checked_items + item;
next_items := QUERY ( z <* bag_to_set(USEDIN(item, '')) | ((
    'CAST_PARTS_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) OR (
    'CAST_PARTS_SCHEMA.FOUNDED_ITEM' IN TYPEOF(z))) );
IF SIZEOF(next_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(next_items) BY 1;
        IF NOT (next_items[i] IN new_check_items) THEN
            result_items := (result_items + next_items[i]) + using_items(
                next_items[i], new_check_items);
        END_IF;
    END_REPEAT;
END_IF;

```

```

    END_REPEAT;
  END_IF;
  RETURN(result_items);

```

```

END_FUNCTION; -- using_items
(*)

```

### **using\_representations**

```

*)
FUNCTION using_representations(
    item: founded_item_select
  ): SET OF representation;

  LOCAL
    results          : SET OF representation;
    intermediate_items : SET OF founded_item_select;
    result_bag       : BAG OF representation;
  END_LOCAL;
  results := [];
  result_bag := USEDIN(item, 'CAST_PARTS_SCHEMA.REPRESENTATION.ITEMS');
  IF SIZEOF(result_bag) > 0 THEN
    REPEAT i := 1 TO HIINDEX(result_bag) BY 1;
      results := results + result_bag[i];
    END_REPEAT;
  END_IF;
  intermediate_items := using_items(item, []);
  IF SIZEOF(intermediate_items) > 0 THEN
    REPEAT i := 1 TO HIINDEX(intermediate_items) BY 1;
      result_bag := USEDIN(intermediate_items[i],
        'CAST_PARTS_SCHEMA.REPRESENTATION.ITEMS');
      IF SIZEOF(result_bag) > 0 THEN
        REPEAT j := 1 TO HIINDEX(result_bag) BY 1;
          results := results + result_bag[j];
        END_REPEAT;
      END_IF;
    END_REPEAT;
  END_IF;
  RETURN(results);

```

```

END_FUNCTION; -- using_representations
(*)

```

### **valid\_calendar\_date**

```

*)
FUNCTION valid_calendar_date(
    date: calendar_date
  ): LOGICAL;
  CASE date.month_component OF
  1 : RETURN((1 <= date.day_component) AND (date.day_component
    <= 31));
  2 : BEGIN
    IF leap_year(date.year_component) THEN
      RETURN((1 <= date.day_component) AND (date.day_component <= 29));
    ELSE
      RETURN((1 <= date.day_component) AND (date.day_component <= 28));
    END_IF;
  END_CASE;

```

```

        END_IF;
    END;
3 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 31));
4 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 30));
5 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 31));
6 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 30));
7 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 31));
8 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 31));
9 :      RETURN((1 <= date.day_component) AND (date.day_component
        <= 30));
10 :     RETURN((1 <= date.day_component) AND (date.
        day_component <= 31));
11 :     RETURN((1 <= date.day_component) AND (date.
        day_component <= 30));
12 :     RETURN((1 <= date.day_component) AND (date.
        day_component <= 31));
    END_CASE;
    RETURN(FALSE);

```

```
END_FUNCTION; -- valid_calendar_date
```

```
(*
```

### **valid\_measure\_value**

```
*)
```

```

FUNCTION valid_measure_value(
    m: measure_value
): BOOLEAN;
IF 'REAL' IN TYPEOF(m) THEN
    RETURN(m > 0);
ELSE
    IF 'INTEGER' IN TYPEOF(m) THEN
        RETURN(m > 0);
    ELSE
        RETURN(TRUE);
    END_IF;
END_IF;

```

```
END_FUNCTION; -- valid_measure_value
```

```
(*
```

### **valid\_time**

```
*)
```

```

FUNCTION valid_time(
    time: local_time
): BOOLEAN;
IF EXISTS(time.second_component) THEN
    RETURN(EXISTS(time.minute_component));
ELSE
    RETURN(TRUE);

```

```

END_IF;

END_FUNCTION; -- valid_time
(*)
valid_units
*)
FUNCTION valid_units(
    m: measure_with_unit
): BOOLEAN;
IF 'CAST_PARTS_SCHEMA.LENGTH_MEASURE' IN TYPEOF(m.value_component)
THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.MASS_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,1,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.TIME_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,1,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.ELECTRIC_CURRENT_MEASURE' IN TYPEOF(m.
value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,1,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.THERMODYNAMIC_TEMPERATURE_MEASURE' IN TYPEOF(m.
value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,1,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.CELSIUS_TEMPERATURE_MEASURE' IN TYPEOF(m.
value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,1,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.AMOUNT_OF_SUBSTANCE_MEASURE' IN TYPEOF(m.
value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,1,0) THEN

```

```

        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.LUMINOUS_INTENSITY_MEASURE' IN TYPEOF(m.
    value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,1) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.PLANE_ANGLE_MEASURE' IN TYPEOF(m.value_component)
    THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.SOLID_ANGLE_MEASURE' IN TYPEOF(m.value_component)
    THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.AREA_MEASURE' IN TYPEOF(m.value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(2,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.VOLUME_MEASURE' IN TYPEOF(m.value_component)
    THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(3,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.RATIO_MEASURE' IN TYPEOF(m.value_component)
    THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.POSITIVE_LENGTH_MEASURE' IN TYPEOF(m.
    value_component) THEN
    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(1,0,0,0,0,0,0) THEN
        RETURN (FALSE);
    END_IF;
END_IF;
IF 'CAST_PARTS_SCHEMA.POSITIVE_PLANE_ANGLE_MEASURE' IN TYPEOF(m.
    value_component) THEN

```



```

    IF derive_dimensional_exponents(m.unit_component) <>
        dimensional_exponents(0,0,0,0,0,0,0,0) THEN
        RETURN(FALSE);
    END_IF;
END_IF;
RETURN(TRUE);

```

```
END_FUNCTION; -- valid_units
```

```
(*
```

## **vector\_difference**

```
*)
```

```

FUNCTION vector_difference(
    arg1, arg2: vector_or_direction
): vector;

LOCAL
    ndim    : INTEGER;
    mag2    : REAL;
    mag1    : REAL;
    mag     : REAL;
    res     : direction;
    vec1    : direction;
    vec2    : direction;
    result  : vector;
END_LOCAL;
IF ((NOT EXISTS(arg1)) OR (NOT EXISTS(arg2))) OR (arg1.dim <> arg2.dim)
    THEN
    RETURN(?);
ELSE
BEGIN
    IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(arg1) THEN
        mag1 := arg1.magnitude;
        vec1 := arg1.orientation;
    ELSE
        mag1 := 1;
        vec1 := arg1;
    END_IF;
    IF 'CAST_PARTS_SCHEMA.VECTOR' IN TYPEOF(arg2) THEN
        mag2 := arg2.magnitude;
        vec2 := arg2.orientation;
    ELSE
        mag2 := 1;
        vec2 := arg2;
    END_IF;
    vec1 := normalise(vec1);
    vec2 := normalise(vec2);
    ndim := SIZEOF(vec1.direction_ratios);
    mag := 0;
    res := dummy_gri || direction(vec1.direction_ratios);
    REPEAT i := 1 TO ndim BY 1;
        res.direction_ratios[i] := (mag1 * vec1.direction_ratios[i]) + (
            mag2 * vec2.direction_ratios[i]);
        mag := mag + (res.direction_ratios[i] * res.direction_ratios[i]);
    END_REPEAT;
END;

```

```

END_REPEAT;
IF mag > 0 THEN
    result := dummy_gri || vector(res,SQRT(mag));
ELSE
    result := dummy_gri || vector(vec1,0);
END_IF;
END;
END_IF;
RETURN(result);

```

```
END_FUNCTION; -- vector_difference
```

(\*

### **get\_property\_definition\_shape\_aspect**

\*)

```

FUNCTION get_property_definition_shape_aspect(
    c_def_instance: characterized_definition
): SET OF shape_aspect;

```

```

LOCAL
    pdr_set : SET OF shape_aspect:= [];
    pd_set  : SET OF property_definition := [];
END_LOCAL;
pd_set := bag_to_set(USEDIN(c_def_instance,
    'CAST_PARTS_SCHEMA.PROPERTY_DEFINITION.DEFINITION'));
IF SIZEOF(pd_set) < 1 THEN
    RETURN(pdr_set);
END_IF;
REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
    pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
        'CAST_PARTS_SCHEMA.SHAPE_ASPECT.OF_SHAPE'));
END_REPEAT;
RETURN(pdr_set);

```

```
END_FUNCTION;
```

(\*

### **get\_action\_property\_representation**

\*)

```

FUNCTION get_action_property_representation(
    c_def_instance: characterized_action_definition
): SET OF action_property_representation;

```

```

LOCAL
    pdr_set : SET OF action_property_representation:= [];
    pd_set  : SET OF action_property := [];
END_LOCAL;
pd_set := bag_to_set(USEDIN(c_def_instance,
    'CAST_PARTS_SCHEMA.ACTION_PROPERTY.DEFINITION'));
IF SIZEOF(pd_set) < 1 THEN
    RETURN(pdr_set);

```

```

    END_IF;
    REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
        pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
'CAST_PARTS_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY'));
    END_REPEAT;
    RETURN(pdr_set);

END_FUNCTION;
(*)
get_resource_property_representation
*)
FUNCTION get_resource_property_representation(
    c_def_instance: characterized_resource_definition
): SET OF resource_property_representation;

LOCAL
    pdr_set : SET OF resource_property_representation:= [];
    pd_set : SET OF resource_property := [];
END_LOCAL;
pd_set := bag_to_set(USEDIN(c_def_instance,
    'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY.RESOURCE'));
IF SIZEOF(pd_set) < 1 THEN
    RETURN(pdr_set);
END_IF;
REPEAT i := 1 TO HIINDEX(pd_set) BY 1;
    pdr_set := pdr_set + bag_to_set(USEDIN(pd_set[i],
'CAST_PARTS_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY'));
    END_REPEAT;
    RETURN(pdr_set);

END_FUNCTION;

END_SCHEMA; -- cast_parts_schema

```

**Annex B****(normative)****AIM short names**

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

**Table B.1 — AIM short names of entities**

AIM Element	Short name
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_METHOD_ASSIGNMENT	ACMTAS
ACTION_METHOD_RELATIONSHIP	ACMTRL
ACTION_METHOD_ROLE	ACM0
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS	AMWAD
ACTION_PROPERTY	ACTPRP
ACTION_PROPERTY_RELATIONSHIP	ACPRRL
ACTION_PROPERTY_REPRESENTATION	ACPRRP
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_RESOURCE	ACTRSR
ACTION_RESOURCE_RELATIONSHIP	ACRSRL
ACTION_RESOURCE_REQUIREMENT	ACRSRQ
ACTION_RESOURCE_TYPE	ACRSTY
ACTION_STATUS	ACTSTT
ACTIVITY_EXECUTION_RESULT	ACEXRS
ACTIVITY_RESULT_RECORD	ACRSRC
ADDRESS	ADDRSS

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ALLOWED_TIME	ALLTM
ALTERNATE_ACTION_METHOD_RELATIONSHIP	AAMR
ALTERNATE_PLAN_RELATIONSHIP	ALPLRL
ANCILLARY_ACTIVITY	ANCACT
ANCILLARY_SETUP	ANCSTP
ANGULAR_LOCATION	ANGLCT
ANGULAR_SIZE	ANGSZ
ANGULARITY_TOLERANCE	ANGTLR
APEX	APEX
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPLIED_ACTION_ASSIGNMENT	APACAS
APPLIED_ACTION_METHOD_ASSIGNMENT	AAMA
APPLIED_ACTION_REQUEST_ASSIGNMENT	AARA
APPLIED_APPROVAL_ASSIGNMENT	APAPAS
APPLIED_AREA	APPAR
APPLIED_CLASSIFICATION_ASSIGNMENT	APCLAS
APPLIED_DATE_AND_TIME_ASSIGNMENT	ADATA
APPLIED_DATE_ASSIGNMENT	APDTAS
APPLIED_DOCUMENT_REFERENCE	APDCRF
APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	ADUCA
APPLIED_EFFECTIVITY_ASSIGNMENT	APEFAS
APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT	AEIA
APPLIED_GROUP_ASSIGNMENT	APGRAS
APPLIED_IDENTIFICATION_ASSIGNMENT	APIDAS
APPLIED_LIBRARY_ASSIGNMENT	APLBAS
APPLIED_ORGANIZATION_ASSIGNMENT	APORAS
APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT	APAOA

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT	ASCA
APPLIED_TIME_ASSIGNMENT	APTMAS
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
APPROVAL_DATE_TIME	APDTTM
APPROVAL_PERSON_ORGANIZATION	APPROR
APPROVAL_ROLE	APPRL
APPROVAL_STATUS	APPSTT
ASSEMBLY	ASSMBL
ASSEMBLY_COMPONENT_USAGE	ASCMUS
AXIS1_PLACEMENT	AX1PLC
AXIS2_PLACEMENT_2D	A2PL2D
AXIS2_PLACEMENT_3D	A2PL3D
B_SPLINE_CURVE	BSPCR
B_SPLINE_CURVE_WITH_KNOTS	BSCWK
B_SPLINE_SURFACE	BSPSR
B_SPLINE_SURFACE_WITH_KNOTS	BSSWK
BANKOUT_FRAME	BNKFRM
BENCH_INSPECTION	BNCINS
BEZIER_CURVE	BZRCRV
BEZIER_SURFACE	BZRSRF
BOSS	BOSS
BOSS_TOP	BSSTP
BOUNDARY_CONDITION	BNDCND
BOUNDED_CURVE	BNDCRV
BOUNDED_SURFACE	BNDSRF
BREP_WITH_VOIDS	BRWTVD
CALENDAR_DATE	CLNDT
CARTESIAN_POINT	CRTPNT
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
CAST_PART	CSTPRT
CAST_PART_WITH_RIGGING	CPWR
CASTING_ACTIVITY	CSTACT
CASTING_DESIGN_FEATURE	CSDSFT
CASTING_ENVELOPE_REQUIREMENT	CSENRQ
CASTING_EQUIPMENT	CSTEQP
CASTING_EQUIPMENT_PROCESS	CSEQPR
CASTING_EQUIPMENT_SETUP	CSEQST
CASTING_FEATURE_DEFINITION	CSFTDF
CASTING_FEATURE_SIZE	CSFTSZ
CASTING_INSERT	CSTINS
CASTING_INSTANCED_FEATURE	CSINFT
CASTING_METHOD	CSTMTH
CASTING_PER_ORDER	CSPROR
CASTING_PROCESS	CSTPRC
CASTING_PRODUCT_DEFINITION	CSPRDF
CASTING_SERVICE_DATA	CSSRDT
CASTING_VERIFICATION	CSTVRF
CENTRE_OF_SYMMETRY	CNOFSY
CHAMFER	CHMFR
CHAMFER_OFFSET	CHMOFF
CHAPLET	CHPLT
CHAPLET_PAD	CHPPD
CHARACTERIZED_OBJECT	CHROBJ
CHECKING_AID_TOOL	CHADTL
CHILL	CHILL
CHOKE	CHOKE
CIRCLE	CIRCLE
CIRCULAR_CLOSED_PROFILE	CRCLPR
CIRCULAR_PATTERN	CRCPTT
CIRCULAR_RUNOUT_TOLERANCE	CRRNTL

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
CLASS	CLASS
CLASSIFICATION_ASSIGNMENT	CLSASS
CLASSIFICATION_ROLE	CLSRL
CLOSED_PATH_PROFILE	CLPTPR
CLOSED_SHELL	CLSSHL
COAXIALITY_TOLERANCE	CXLTLR
COMMON_DATUM	CMMDTM
COMPOSITE_CURVE	CMPCRIV
COMPOSITE_CURVE_ON_SURFACE	CCOS
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_HOLE	CMPHL
COMPOSITE_SHAPE_ASPECT	CMSHAS
COMPOSITION_ELEMENT	CMPELM
COMPOSITION_ELEMENT_RECORD	CMELRG
COMPOSITION_INSPECTION	CMPINS
COMPOUND_FEATURE	CMPTFR
COMPOUND_REPRESENTATION_ITEM	CMRPIT
CONCENTRICITY_TOLERANCE	CNCTLR
CONDITION_OR_ASSUMPTION	CNORAS
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_FACE_SET	CNFCST
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONTINUOUS_PROCESS_RELATIONSHIP	CNPRRL
CONTROLLER	CNTRLL
CONVERSION_BASED_UNIT	CNBSUN
COOLING_PORT	CLNPRT
COORDINATED_UNIVERSAL_TIME_OFFSET	CUTO
CORE	CORE
CORE_ASSEMBLY	CRASS
CORE_BENCH	CRBNC



**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
CORE_BOX_TOOLING	CRBXTL
CORE_BOX_VENT	CRBXVN
CORE_CONVEYER_SYSTEM	CRCNSY
CORE_EQUIPMENT	CREQP
CORE_MACHINE	CRMCH
CORE_MASTER	CRMST
CORE_PRINT	CRPRN
CORE_SETTER	CRSTT
CORROSION_SERVICE	CRRSRV
CURVE	CURVE
CUSTOMER_CASTING_REQUIREMENT	CSCSRQ
CUSTOMER_SIMULATION	CSTSMML
CYLINDRICAL_SURFACE	CYLSRF
CYLINDRICITY_TOLERANCE	CYLTLR
DATA_CURVE	DTCRV
DATA_ENVIRONMENT	DTENV
DATE	DATE
DATE_AND_TIME	DTANTM
DATE_AND_TIME_ASSIGNMENT	DATA
DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATE_TIME_ROLE	DTMRL
DATED_EFFECTIVITY	DTDEFF
DATUM	DATUM
DATUM_FEATURE	DTMFTR
DATUM_REFERENCE	DTMRFR
DATUM_TARGET	DTMTRG
DEFECT_PREDICTION	DFCPRD
DEFINING_ACTION_METHOD_RELATIONSHIP	DAMR
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
DERIVED_SHAPE_ASPECT	DRSHAS
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTION_ATTRIBUTE	DSCATT
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
DESIGN_PART	DSGPRT
DESIGN_REFERENCE	DSGRFR
DIE_CAST_MASTER	DCSMS
DIE_CAST_TOOLING	DCSTL
DIE_CLAMPING	DCLM
DIE_CORE	DCR
DIE_DESIGN_FEATURE	DDSFT
DIE_MASTER	DMST
DIE_MOULD_TOOLING	DMLTL
DIE_MOULD_VENT	DMLVN
DIMENSION_RELATED_TOLERANCE_ZONE_ELEMENT	DRTZE
DIMENSIONAL_CHARACTERISTIC_REPRESENTATION	DMCHRP
DIMENSIONAL_EXPONENTS	DMNEXP
DIMENSIONAL_LOCATION	DMNLCT
DIMENSIONAL_LOCATION_WITH_PATH	DLWP
DIMENSIONAL_MEASUREMENT_INSPECTION	DMMSIN
DIMENSIONAL_SIZE	DMNSZ
DIMENSIONAL_SIZE_WITH_PATH	DSWP
DIRECTED_ACTION	DRCACT
DIRECTED_DIMENSIONAL_LOCATION	DRDMLC
DIRECTION	DRCTN
DIRECTION_SHAPE_REPRESENTATION	DRSHRP
DOCUMENT	DCMNT
DOCUMENT_FILE	DCMFL
DOCUMENT_FILE_PROPERTIES	DCFLPR
DOCUMENT_PRODUCT_ASSOCIATION	DCPI

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
DOCUMENT_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_REPRESENTATION_TYPE	DCRPTY
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN
DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	DUCA
DOCUMENT_USAGE_ROLE	DCUSRL
DRAFTED_SURFACE	DRFSRF
DRYER	DRYER
EDGE	EDGE
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EDGE_ROUND	EDGRND
EDGE_SHAPE_REPRESENTATION	EDSHRP
EFFECTIVITY	EFFCTV
EFFECTIVITY_ASSIGNMENT	EFFASS
EJECTOR_BOX	EJCBX
EJECTOR_DESIGN_FEATURE	EJDSFT
EJECTOR_EQUIPMENT	EJCEQP
EJECTOR_PIN	EJCPN
EJECTOR_PLATE	EJCPLT
EJECTOR_SYSTEM	EJCSYS
ELEMENTARY_SURFACE	ELMSRF
ELEVATED_TEMPERATURE_SERVICE	ELTMSR
ELLIPSE	ELLPS
EQUIPMENT_SETTING_RECORD	EQSTRC
EQUIPMENT_USAGE	EQPUSG
EXECUTED_ACTION	EXCACT
EXTENSION	EXTNSN
EXTERNAL_IDENTIFICATION_ASSIGNMENT	EXIDAS
EXTERNAL_SOURCE	EXTSRC

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
EXTERNALLY_DEFINED_CLASS	EXD0
EXTERNALLY_DEFINED_DIMENSION_DEFINITION	EDDD
EXTERNALLY_DEFINED_FEATURE_DEFINITION	EDFD
EXTERNALLY_DEFINED_GENERAL_PROPERTY	EDGP
EXTERNALLY_DEFINED_ITEM	EXDFIT
EXTERNALLY_DEFINED_ITEM_RELATIONSHIP	EDIR
EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS	EDRWP
EXTERNALLY_DEFINED_SCHEMA	EXDFSC
FABRICATION	FBRCTN
FACE	FACE
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SHAPE_REPRESENTATION	FCSHRP
FACE_SHAPE_REPRESENTATION_RELATIONSHIP	FSRR
FACE_SURFACE	FCSRF
FACETED_BREP	FCTBR
FEATURE_COMPONENT_DEFINITION	FTCMDF
FEATURE_COMPONENT_RELATIONSHIP	FTCMRL
FEATURE_DEFINITION	FTRDFN
FEATURE_PATTERN	FTRPTT
FEEDER	FEEDER
FILLET	FILLET
FILTER	FILTER
FINISHING_AND_MACHINING_OPERATION	FAMO
FINISHING_EQUIPMENT	FNSEQP
FIRST_ARTICLE	FRSART
FIXTURE_TAB	FXTTB
FLASK	FLASK
FLASKLESS	FLSKLS
FLAT_FACE	FLTFC
FLATNESS_TOLERANCE	FLTTLR

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
FOUNDED_ITEM	FNDITM
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
GATING_DESIGN_FEATURE	GTDSFT
GATING_SYSTEM	GTNSYS
GEAR	GEAR
GENERAL_PROPERTY	GNRPRP
GENERAL_PROPERTY_ASSOCIATION	GNPRAS
GENERIC_MANUFACTURING_RESOURCE	GNMNRS
GEOMETRIC_ALIGNMENT	GMTALG
GEOMETRIC_INTERSECTION	GMTINT
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_TOLERANCE	GMTTLR
GEOMETRIC_TOLERANCE_RELATIONSHIP	GMTLRL
GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE	GTWDR
GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT	GTWDU
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUP_ASSIGNMENT	GRPASS
GROUP_RELATIONSHIP	GRPRLT
HEAT_TREAT	HTTRT
HEAT_TREAT_REQUIREMENT	HTTRRQ
HOLE_BOTTOM	HLBTT
HYPERBOLA	HYPRBL
ID_ATTRIBUTE	IDATT
IDENTIFICATION_ASSIGNMENT	IDNASS
IDENTIFICATION_ROLE	IDNRL
ILLUSTRATION	ILLSTR
IMPREGNATION_ACTIVITY	IMPACT
IN_FACILITY_LOCATION	INFCLC

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
INGATE	INGATE
INGATE_CONTACT	INGCNT
INSPECTION_ACTIVITY	INSACT
INSPECTION_OR_TEST_REQUIREMENT	IOTR
INSPECTION_OR_TEST_RESULT	IOTO
INSPECTION_PLAN	INSPLN
INSTANCED_FEATURE	INSFTR
INTEGRATION_INTERVAL	INTINT
INVESTMENT_CASTING_MASTER	INCSMS
INVESTMENT_CASTING_TOOLING	INCSTL
INVESTMENT_CASTING_TOOLING_DEFINITION	ICTD
INVESTMENT_DESIGN_FEATURE	INDSFT
INVESTMENT_MOULD	INVMLD
KNOWN_SOURCE	KNWSRC
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIBRARY_CLASS_VERSION_ASSIGNMENT	LCVA
LIBRARY_PROPERTY_VERSION_ASSIGNMENT	LPVA
LIMITS_AND_FITS	LMANFT
LINE	LINE
LINE_PROFILE_TOLERANCE	LNP0
LINEAR_PROFILE	LNRPRF
LOCAL_TIME	LCLTM
LOCATION_SHAPE_REPRESENTATION	LCSHRP
LOOP	LOOP
LOOSE_PIECE	LSPC
LOST_FOAM_CASTING_DIE	LFCD
LOT_EFFECTIVITY	LTEFF
MACHINE	MCHN
MACHINE_ALLOWANCE	MCHALL
MACHINE_ELEMENT_RELATIONSHIP	MCELRL

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
MACHINE_MOUNTING	MCHMNT
MACHINE_SETUP	MCHSTP
MACHINING_ACTIVITY	MCHACT
MACHINING_PROCESS	MCHPRC
MAKE_FROM_USAGE_OPTION	MFUO
MANIFOLD_SOLID_BREP	MNSLBR
MANUFACTURING_ACTIVITY	MNFACT
MANUFACTURING_ACTIVITY_RELATIONSHIP	MNACRL
MANUFACTURING_PROCESS	MNFPRC
MANUFACTURING_PROCESS_RELATIONSHIP	MNPRRL
MAPPED_ITEM	MPPITM
MARKING	MRKNG
MASS_MEASURE_WITH_UNIT	MMWU
MASS_UNIT	MSSUNT
MASTER_PATTERN	MSTPTT
MASTER_SAMPLE	MSTSMP
MATERIAL_DESIGNATION	MTRDSG
MATERIAL_PROPERTY	MTRPRP
MATERIAL_PROPERTY_REPRESENTATION	MTPRRP
MATERIAL_STRUCTURE	MTRSTR
MATERIAL_USAGE	MTRUSG
MATERIAL_USAGE_RECORD	MTUSRC
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MECHANICAL_PROPERTY_REQUIREMENTS	MCPRRQ
MECHANICAL_STRESS	MCH2
MELTING_EQUIPMENT	MLTEQP
MESHING_CONDITION	MSHCND
METALCASTER_SIMULATION	MTLSML
MISMATCH_TOLERANCE	MSMTLR

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
MODIFIED_GEOMETRIC_TOLERANCE	MDGMTL
MODIFIED_PATTERN	MDFPTT
MOULD_BOX	MLDBX
MOULD_LOCK	MLDLCK
MOULDING_EQUIPMENT	MLDEQP
NAME_ATTRIBUTE	NMATT
NAMED_UNIT	NMDUNT
NEXT_ASSEMBLY_USAGE_OCCURRENCE	NAUO
NGON_CLOSED_PROFILE	NGCLPR
NON_CASTING_ACTIVITY	NNCSAC
NON_CASTING_EQUIPMENT_PROCESS	NCEP
NON_MACHINING_PROCESS	NNMCPR
NON_PERMANENT_MOULDING_PROCESS	NPMP
OBJECT_ROLE	OBJRL
OPEN_PATH_PROFILE	OPPTPR
OPEN_SHELL	OPNSHL
ORDERED_PART	ORDPRT
ORGANIZATION	ORGNZT
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_ROLE	ORGRL
ORGANIZATIONAL_ADDRESS	ORGADD
ORGANIZATIONAL_PROJECT	ORGPRJ
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
ORIENTED_SURFACE	ORNSRF
OTHER_INSPECTION	OTHINS
OUTER_ROUND	OTRRND
OUTSIDE_PROFILE	OTSPRF



**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
PARABOLA	PRBL
PARALLEL_OFFSET	PRLOFF
PARALLELISM_TOLERANCE	PRLTLR
PARAMETRIC_REPRESENTATION_CONTEXT	PRRPCN
PARTIAL_CIRCULAR_PROFILE	PRCRPR
PARTING_SURFACE	PRTSRF
PATH	PATH
PATH_FEATURE_COMPONENT	PTFTCM
PATH_SHAPE_REPRESENTATION	PTSHRP
PATTERN_EQUIPMENT_AVAILABLE	PTEQAV
PATTERN_OFFSET_MEMBERSHIP	PTOFMM
PATTERN_OMIT_MEMBERSHIP	PTOMMM
PATTERN_PLATE	PTTPLT
PAYMENT_AND_SHIPPING	PYANSH
PCURVE	PCURVE
PERMANENT_MOULDING_PROCESS	PRMLPR
PERPENDICULAR_TO	PRPT
PERPENDICULARITY_TOLERANCE	PRPTLR
PERSON	PERSON
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSONAL_ADDRESS	PRSADD
PIN_CENTER	PNCNT
PIN_TIP	PNTTP
PLACED_DATUM_TARGET_FEATURE	PDT0
PLACEMENT	PLCMNT
PLANAR_SHAPE_REPRESENTATION	PLSHRP
PLANE	PLANE
PLANE_ANGLE_MEASURE_WITH_UNIT	PAMWU
PLANE_ANGLE_UNIT	PLANUN

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
PLUS_MINUS_TOLERANCE	PLMNTL
POCKET	POCKET
POCKET_BOTTOM	PCKBTT
POINT	POINT
POLY_LOOP	PLYLP
POLYLINE	PLYLN
POSITION_TOLERANCE	PSTTLR
PRE_DEFINED_ITEM	PRDFIT
PRECISION_QUALIFIER	PRCQLF
PROCESS_PARAMETER_RECORD	PRPRRC
PROCESS_PLAN_ACTIVITY	PRPLAC
PROCESS_PLAN_SECURITY	PRPLSC
PROCESS_PLAN_SPECIFICATION	PRPLSP
PROCESS_PLAN_VERSION	PRPLVR
PROCESS_PRODUCT_ASSOCIATION	PRPRAS
PROCESS_PROPERTY_ASSOCIATION	PRPRS
PROCESS_REQUIREMENT	PRCRQR
PRODUCT	PRDCT
PRODUCT_CATEGORY	PRDCTG
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_EFFECTIVITY	PRDFEF
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_FORMATION_RELATIONSHIP	PDFR
PRODUCT_DEFINITION_PROCESS	PRDFPR
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_USAGE	PRDFUS
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD
PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP	PMCR

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
PRODUCT_RELATED_PRODUCT_CATEGORY	PRPC
PRODUCTION_CORE_BOX	PRCRBX
PRODUCTION_DIE_CAST_MOULD	PDCM
PRODUCTION_DIE_MOULD	PRDML
PRODUCTION_INVESTMENT_CAST_MOULD	PICM
PRODUCTION_PATTERN_DEFINITION	PRPTDF
PRODUCTION_RATE	PRDRT
PRODUCTION_TOOL	PRDTL
PRODUCTION_WELDING	PRDWLD
PROFILE_FLOOR	PRFFLR
PROJECTED_ZONE_DEFINITION	PRZNDF
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
PROPERTY_INSPECTION	PRPINS
PROPERTY_PROCESS	PRPPRC
PROPERTY_RELATIONSHIP	PRPRLT
PROTRUSION	PRTRSN
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS
RECTANGULAR_CLOSED_PROFILE	RCCLPR
RECTANGULAR_PATTERN	RCTPTT
RED_LINED_PART	RDLNPR
REFERENCED_MODIFIED_DATUM	RFMDDT
RELATIONSHIP_CONDITION	RLTCND
REMOVAL_VOLUME	RMVVLN

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
REPLICATE_FEATURE	RPLFTR
REPORTING_REQUIREMENT	RPRRQR
REPRESENTATION	RPRSNT
REPRESENTATION_CONTEXT	RPRCNT
REPRESENTATION_ITEM	RPRITM
REPRESENTATION_MAP	RPRMP
REPRESENTATION_RELATIONSHIP	RPRRLT
REQUEST_FOR_QUOTATION	RQFRQT
REQUIREMENT_FOR_ACTION_RESOURCE	RFAR
RESET_OR_PUSH_PIN	ROPP
RESOURCE_PROPERTY	RSRPRP
RESOURCE_PROPERTY_REPRESENTATION	RSPRRP
RESOURCE_REQUIREMENT_TYPE	RSRQTY
RESOURCE_WITH_MATERIAL	RSWTMT
RESOURCE_WITH_REPRESENTATION	RSWTRP
REVISION	RVSN
REVOLVED_PROFILE	RVLPRF
RIB	RIB
RIB_TOP	RBTP
RIB_TOP_FLOOR	RBTPFL
RISER	RISER
RISER_CONTACT	RSRCNT
ROLE_ASSOCIATION	RLASS
ROUND_HOLE	RNDHL
ROUNDED_END	RNDEND
ROUNDED_U_PROFILE	RNUPR
ROUNDNESS_TOLERANCE	RNDTLR
RUNNER	RUNNER
RUNOUT_ZONE_DEFINITION	RNZNDF
RUNOUT_ZONE_ORIENTATION	RNZNOR
RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION	RZORD

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
SAMPLED_SET	SMPST
SAND_CAST_DESIGN_FEATURE	SCDF
SAND_CASTING_TOOLING	SNCSTL
SAND_MOULD	SNDMLD
SECONDARY_TOOLING	SCNTLN
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SEQUENTIAL_METHOD	SQNMTH
SERIAL_ACTION_METHOD	SRACMT
SHAKEOUT	SHKT
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_DERIVING_RELATIONSHIP	SADR
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_DEFINING_RELATIONSHIP	SHDFRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_DIMENSION	SHPDMN
SHAPE_DIMENSION_REPRESENTATION	SHDMRP
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_WITH_PARAMETERS	SRWP
SHOT_SLEEVE	SHTSLV
SI_UNIT	SUNT
SIMULATED_PROPERTY	SMLPRP
SIMULATION_INPUT	SMLINP
SIMULATION_INPUT_REGION	SMINRG
SIMULATION_OUTPUT	SMLOTP
SIMULATION_OUTPUT_REGION	SMOTRG
SIMULATION_PROCESS	SMLPRC
SIMULATION_RESULT	SMLRSL
SIMULATION_RUN	SMLRN
SIMULATION_RUN_RELATIONSHIP	SMRNRL

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
SIMULATION_SOFTWARE	SMLSFT
SIMULATION_UNIT	SMLUNT
SIMULATION_UNIT_STATE	SMUNST
SINGLE_ACTIVITY_RELATIONSHIP	SNACRL
SLIDE	SLIDE
SLOT	SLOT
SLOT_END	SLTEND
SOLID_ANGLE_UNIT	SLANUN
SOLID_MODEL	SLDMDL
SPECIAL_INSPECTION_REQUIREMENT	SPINRQ
SPHERICAL_CAP	SPHCP
SPHERICAL_SURFACE	SPHSRF
SPRUE	SPRUE
SPRUE_AND_RUNNER_MOULD	SARM
SQUARE_U_PROFILE	SQUPR
STANDARD_UNCERTAINTY	STNUNC
STEP	STEP
STOP_PIN	STPPN
STRAIGHTNESS_TOLERANCE	STRTLR
SURFACE	SRFC
SURFACE_CURVE	SRFCRV
SURFACE_FINISH	SRFFNS
SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PROFILE_TOLERANCE	SRPRTL
SWEPT_SURFACE	SWPSRF
SYMMETRIC_SHAPE_ASPECT	SYSHAS
SYMMETRY_TOLERANCE	SYMTLR
TANGENT	TNGNT
TAPER	TAPER
TEE_PROFILE	TPRF

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
THREAD	THREAD
TIME_ASSIGNMENT	TMASS
TIME_MEASURE_WITH_UNIT	TMWU
TIME_ROLE	TMRL
TIME_UNIT	TMUNT
TOLERANCE_INSPECTION	TLRINS
TOLERANCE_REQUIREMENT	TLRRQR
TOLERANCE_VALUE	TLRVL
TOLERANCE_ZONE	TLRZN
TOLERANCE_ZONE_DEFINITION	TLZNDF
TOLERANCE_ZONE_FORM	TLZNFR
TOOL_VERIFICATION	TLVRF
TOOLING_PROCESS	TLNPRC
TOPOLOGICAL_REPRESENTATION_ITEM	TPRPIT
TOROIDAL_SURFACE	TRDSRF
TOTAL_RUNOUT_TOLERANCE	TTRNTL
TRANSITION_FEATURE	TRNFTR
TURNED_KNURL	TRNKNR
TYPE_QUALIFIER	TYPQLF
UNCERTAINTY_MEASURE_WITH_UNIT	UMWU
UNCERTAINTY_QUALIFIER	UNCQLF
UNIFORM_CURVE	UNFCRV
UNIFORM_SURFACE	UNFSRF
VALUE_RANGE	VLRNG
VALUE_REPRESENTATION_ITEM	VLRPIT
VECTOR	VECTOR
VEE_PROFILE	VPRF
VENT	VENT
VERSIONED_ACTION_REQUEST	VRACRQ
VERTEX	VERTEX
VERTEX_LOOP	VRTLP

**Table B.1 — AIM short names of entities (continued)**

AIM Element	Short name
VERTEX_POINT	VRTPNT
VIEW_REFERENCE	VWRFR
WELL	WELL



## **Annex C**

### **(normative)**

### **Implementation method specific requirements**

The implementation method defines what type of exchange behavior is required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and annotated listing defined in Annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'cast\_parts\_schema'.

**Annex D**  
**(normative)**

**Protocol Implementation Conformance Statement (proforma)**

This clause lists the optional elements of this part of ISO 10303. An implementation may choose to support any combination of these optional elements. However, certain combinations of options are likely to be implemented together. These combinations are called conformance classes and are described in the subclauses of this Annex.

This Annex is in the form of a questionnaire. This questionnaire is intended to be filled out by the implementer and may be used in preparation for conformance testing by a testing laboratory. The completed PICS proforma is referred to as a PICS.

A number of options are identified in this standard for possible use by conforming implementations. Some of these options may be dynamically (run-time) selected for use/non-use, for instance, OPTIONAL attributes of an entity. Others shall be statically (configuration-time) selected for use/non-use, such as a particular style of geometry as defined in a conformance class.

Questions:

Question:

1. Please provide an identifier for the product or system for which conformance is claimed:

Product name and current version number: \_\_\_\_\_

2. Please indicate the implementation method chosen:

ISO 10303-21 exchange structure — preprocessor

Preprocessor name and current version number: \_\_\_\_\_

ISO 10303-21 exchange structure — postprocessor

Postprocessor name and current version number: \_\_\_\_\_

3. There are nine classes defined in this international standard. Each class specifies a subset of this part of ISO 10303 AIM constructs. These classes are detailed in 6 of this document. Conformance to this part of ISO 10303 requires conformance to at least one of the primary conformance classes 1 through 4.

Claimed classes of conformance (functionality) - circle choices:

— Class 1: Minimum for all conformance classes;

— Class 2: Customer to metal caster;

- Class 3: Metal caster to inspection;
- Class 4: Metal caster to simulation;
- Class 5: Metal caster to tooling shop;
- Class 6: Metal caster to other operations;
- Class 7: Customer to simulation;
- Class 8: Quality assurance;
- Class 9: Casting process planning.

4. If the implementation receives data, which does not comply with the requirements in this International Standard for the selected conformance class(es), or with the requirements of the 20 series of parts for the selected implementation method, it shall execute a default response. A default response shall be statically set.

Default Response: \_\_\_\_\_

5. A conforming implementation shall maintain the static options selected throughout subsequent dynamic assessment (testing) without requiring modification. In a user environment, a conforming implementation shall permanently maintain the provision of selected static options, or it shall provide users discretionary control over the changing and setting of the static options, or both (depending on the option).

Does the IUT provide some user discretion over the changing and setting of static options?  
Yes or No

6. If yes, which ones?

(a) Conformance class(es): \_\_\_\_\_

(b) Default Response: \_\_\_\_\_

7. A statement of conformance shall include identification of at least one party deeming conformance for the implementation.

**Evaluator(s) (tester/certifier/accreditor):** \_\_\_\_\_

## **Annex E**

### **(normative)**

#### **Information object registration**

##### **E.1 Document identification**

To provide for unambiguous identification of an information object in an open system, the object identifier

{ iso standard 10303 part(223) version (0) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and described in ISO 10303-1.

##### **E.2 Schema identification**

To provide for unambiguous identification of the schema specification given in this part of ISO 10303 feature-based-process-planning-schema in the open information system, the object identifier are assigned as follows:

{ iso standard 10303 part(223) version (0) object (1) cast-parts-schema(1) }

is assigned to the cast\_parts\_schema (see Annex A).

{ iso standard 10303 part(223) version (0) object (1) cast\_parts\_schema(2) }

is assigned to the cast\_parts\_schema short form schema (see 5.2).

The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

## Annex F

### (informative)

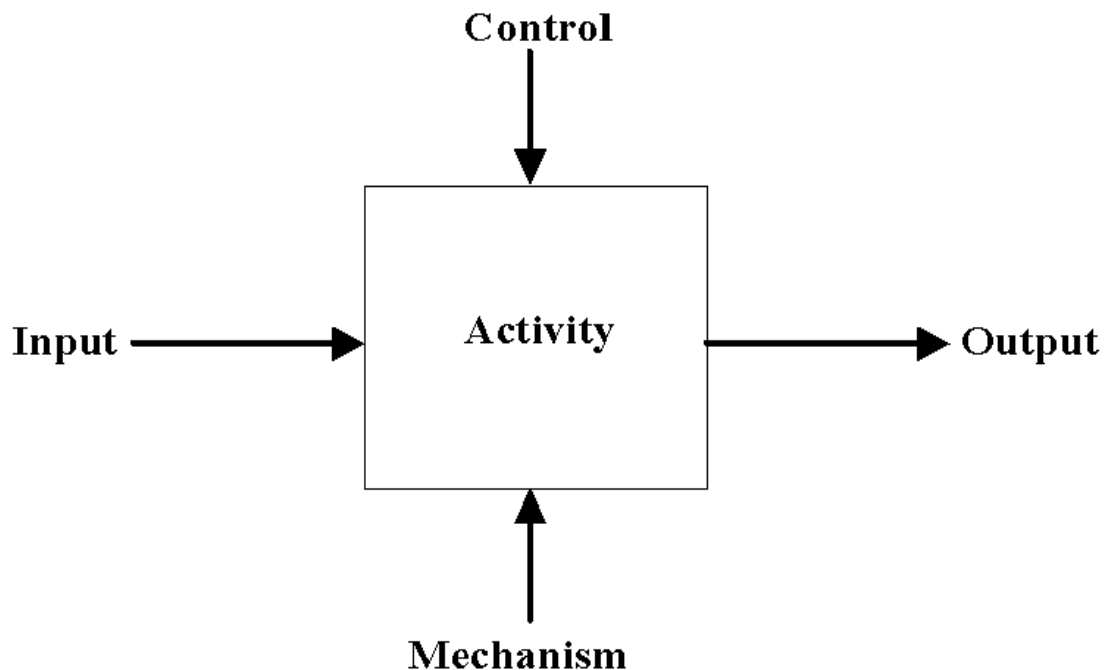
#### Application activity model

The application activity model (AAM) is provided to aid in the understanding the scope and information requirements defined in this part of ISO 10303. The model is presented as a set of activity figures that contain the activity diagrams and a set of definitions of the activities and their data.

The AAM covers activities which go beyond the subject of this part of ISO 10303. The diagrams use a modified IDEF0 notation [2]. Figure F.1 gives the basic notation. Each activity may be decomposed to provide more detail. If an activity has been decomposed, a separate figure is included.

As with any IDEF0 model, the application activity model is dependent on a particular viewpoint and purpose. The viewpoint of the application activity model is from a manufacturing engineer. The purpose of the application activity model is to clarify the context and scope of this part of ISO 10303.

This is an activity model of an enterprise for manufacturing a part. There are several activity diagrams that have all activities out of scope but they are important in illustrating how the manufacture of a part process was developed and how the in-scope requirements were derived.



**Figure F.1 — IDEF0 basic notation**

## F.1 Application activity model definitions

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this part of ISO 10303.

The definitions in this annex do not supersede the definitions given in the main body of the text.

### F.1.1 Application activity model activity definitions

The following terms are used to define the activities of the AAM diagrams.

**F.1.1.1 Cast a part (A0):** all activities and materials required to produce a casting.

**F.1.1.2 Casting machine capabilities:** characteristics of the available equipment that the casting processes shall accommodate in order to make castings.

**F.1.1.3 Casting part description:** all information needed to describe and characterize the part or component which is to be cast, including the complete geometry of the component and relevant attributes, such as dimensional tolerances and any specified surface finishes, the mechanical properties required in the casting, the alloy to be used, and regulations, standards and specifications which apply. Information may also be included about special finishing or testing procedures required by the customer. If a sampling is required prior to commencement of full scale production, these requirements are also specified. Included are detailed descriptions of two items: the part originally by the customer, and a description of the casting part. The geometry of these two parts usually differ slightly. For example, the casting part would contain machining allowances.

**F.1.1.4 Casting process data\*:** the information about the processing methods and techniques used to produce the casting. It includes sand casting process data, die casting process data, investment casting process data, and finishing and inspection process data

**F.1.1.5 Casting for repair:** a casting which has failed inspection tests, and is sent to be reworked or modified, so that it may afterwards be acceptable to the customer.

NOTE Impregnation of castings which shall be pressure tight, or welding of cracks are examples of actions which may be taken to repair castings.

**F.1.1.6 Charge materials and additives for casting\*:** all items used to produce the liquid metal used in casting, including ingot, recycled gates and risers, fluxes, alloying additives charged to the melting furnace, grain refiners, modifiers or inoculants, hardeners or other master alloys and fluxing salts or gases.

**F.1.1.7 Core machine or core box:** the core boxes and related equipment used to produce cores.

**F.1.1.8 Core materials:** materials used to produce cores, including bonded sand mixes, plaster in some applications, and steel or iron in die casting.

**F.1.1.9 Core processing data:** information related to the manufacture of cores, including the composition of materials used, the core box temperature and the holding time for production of shell cores, the curing cycle employed for bonded sand cores, the moisture content of the cores, and the results of any mechanical tests on cores.

NOTE The composition of materials and the results of the mechanical test on the core are in-scope of this AP, all others are out-of-scope.

**F.1.1.10 Cores:** a component added to the mould or die, which is used to produce undercuts, hollow sections or protrusions.

NOTE Cores for low pressure casting processes are usually made from bonded sand, and are produced in a core box or core machine. A core for high pressure die casting is either fixed or movable, and is made from steel or cast iron. A movable core is usually called a slide. It slides into the mould cavity before pouring the shot, and is removed from the casting before the casting is extracted from the mould.

**F.1.1.11 Drawing tools\*:** hardware and software used in the design process to characterize the geometry, shape and size of the parts, or components, under consideration.

**F.1.1.12 Finish (A41):** a raw casting is given one or more treatments in order to prepare it to meet the customer's specifications and requirements. The activities included in finishing are: trimming or cutting gates and risers from the casting, grinding off flash and other unwanted material, shot blasting or other surface finishing techniques, heat treatment, hipping, machining, cleaning, plating and painting.

**F.1.1.13 Finish and inspect (A4):** a sequence of activities required to finish a raw casting for the customer, including trimming, machining, heat treating and inspection.

**F.1.1.14 Finished casting:** a raw casting that has received all necessary finishing operations, but which has not received final inspection and testing.

**F.1.1.15 Finishing and inspection process data:** information that characterizes the process parameters used in finishing operations; and the performance of the castings at each finishing operation; as well as the results of all test and inspection procedures used to evaluate the castings. The process data for repaired castings is also included.

Note Finishing and inspection process data would also include data for welding, fabrication, assembly, surface finish, heat treat, machining activities, or other operations performed for finishing the part.

**F.1.1.16 Finishing equipment and heat treat furnaces\*:** cutting and grinding equipment used to trim or remove rigging and flashing from the casting, equipment used to grind, machine or surface-condition castings, and furnaces used to heat treat or HIP the castings. Items such as quench tanks, heat treating fixtures and baskets are also included.

**F.1.1.17 Finishing process data\*:** information that characterizes the process parameters used in finishing operations, and the performance of the castings at each finishing operation. Included here are data on the heat treatment cycles and hipping procedures employed, and the number of castings scrapped at each finishing stage.

**F.1.1.18 Foundry process specifications:** in-house, or foundry, specifications used to characterize and regulate production processes employed in casting.

**F.1.1.19 Furnace system\*:** equipment used to produce the high temperatures required to melt metal prior to casting, or to produce the heat needed to melt out wax patterns in the de-waxing process.

**F.1.1.20 Furnace system, crucibles, ladles, pumps, cranes and robots\*:** equipment used to melt molten metal, prepare it for casting, and transfer the molten metal into the prepared sand mould, investment mould, or die.

**F.1.1.21 Inspect and Test (A42):** subject the finished castings to a final test and inspection procedure specified for the casting.

EXAMPLE inspection procedures include visual examinations for surface cracks and porosity; radiographic examinations for inclusions, porosity or shrinkage; ultrasonic inspection for internal defects; and checking that the casting meets the dimensional specifications. Examples of testing procedures are measurements of tensile or other mechanical properties, hardness measurements, breaking tests and checks for pressure tightness.

**F.1.1.22 Inspected casting:** the cast part, that has received the desired finishing operation, has been inspected, and has been found to meet the customer's requirements. The cast part is ready to be shipped to the customer from the foundry.

**F.1.1.23 Inspection equipment\*:** tools and machinery used to conduct inspection procedures.

**F.1.1.24 Inspection process data:** quality control process data that records the results of inspection and testing procedures that are performed on the finished casting, prior to shipping the casting to the customer.

NOTE The information included would be: results of visual inspection, compliance with dimensional tolerances, measurement of tensile properties, hardness measurements, microscopic examination, x-ray or other radiographic test results, measurement of pressure tightness, results of ultrasonic examination, and results of fluoroscopic or die penetrant tests for surface cracks and porosity.

**F.1.1.25 Machine tools\*:** equipment used to cut, drill, mill, grind and otherwise shape components in the casting process.

**F.1.1.26 Machine tools, moulding machines, core machines or core boxes:** equipment used in production of sand moulds and cores, and in machining of castings or die components.

NOTE The physical machine tools, moulding machines and core machines are out-of-scope. The identifying characteristics of these machines and core boxes are in-scope.

**F.1.1.27 Material specifications:** items which describe the material under consideration, including its name or trade name, supplier or manufacturer, information related to its composition, and regulations, standards and specifications for this material.



**F.1.1.28 Melt processing data and metal composition\*:** quality control process data which characterizes the melting and melt treatment processes used to prepare liquid metal for casting, and the final resulting composition of the molten alloy.

NOTE 1 Items that are out-of-scope are; charges to the furnace, furnace melting parameters, melt treatment parameters, any ladle treatments employed, and the results of any tests, including thermal analysis and measurements of inclusion or gas contents.

NOTE 2 Items that are in-scope; would include the heat number, a list of materials and weights of materials in the melt.

**F.1.1.29 Mould temperature control system\*:** system used to bring the die casting mould, or the investment casting mould, to the desired temperature prior to pouring liquid metal into the mould

**F.1.1.30 Molten metal:** prepared molten metal, having the proper composition and temperature, before it is introduced into the mould.

**F.1.1.31 Packaged casting:** a casting that has been finished and inspected, and that has been packaged in the desired manner, and is ready for shipment to the customer.

**F.1.1.32 Package (A44):** place of the finished and inspected cast part into the desired shipping container or containers, and weigh the castings before shipment.

NOTE In some cases the customer specifies a procedure for packaging, or even the containers to be used. At times, finished castings are placed into specially designed, pre-moulded skids or shipping containers which may be provided by the customer or by the foundry.

**F.1.1.33 Packaging data:** information related to the act of packaging, such as the number of finished cast parts, or the weight to be shipped.

**F.1.1.34 Process modeling system:** a computer-based simulation of the mould filling and metal freezing processes which occur during casting.

**F.1.1.35 Process plan\*:** a specified sequence of process operations, together with specifications regarding important process operating parameters to be used in each process step, which can be used to produce a single casting, or by repetition, a large quantity of castings. Included in the process plan would be any special instruction, as well as machine settings to be used in a process step.

**F.1.1.36 Quality assurance data:** detailed test and inspection results obtained from a sample lot of castings. These QA results are used to qualify the mould or die design for dimensional correctness, and to qualify the process plan as being capable of producing castings of sufficiently high quality.

**F.1.1.37 Raw casting:** the casting after it is removed from the mould, and before any trimming, grinding, finishing, machining, heat treating or inspection.

**F.1.1.38 Regulations, standards and specifications:** detailed and standardized specifications established by international, national, or trade organizations which apply to the material or process under consideration.

ISO 10303-223:2008 (E)

**F.1.1.39 Repair (A43):** remove defects in castings, and restore the functionality of the cast part.

EXAMPLE repair operations are welding and impregnation of castings that shall be pressure tight.

**F.1.1.40 Repair process data:** quality control process data that is used to characterize the repair procedures used for any particular casting.

**F.1.1.41 Repaired casting:** a defective casting that has received remedial action, and consequently is suitable for the customer's application.

**F.1.1.42 Scrap to recycle:** trim and scrapped castings that will be re-melted in order to recover the contained metal.

**F.1.1.43 Scrapped castings:** castings that do not to meet the customer's specifications and requirements, and that cannot be repaired.

**F.1.1.44 Secondary tooling design:** engineering specifications for any of a number of secondary tooling devices used in core place, finishing, inspection, and packaging, when castings are produced in large quantities.

EXAMPLE devices would include extraction grippers and die spray nozzles in die casting, dies or fixtures for trim or straightening presses, machining fixtures, plating fixtures, finishing fixtures, specially designed shipping containers, gauging fixtures, heat treating fixtures, core removal fixtures, painting fixtures, and leak testing fixtures.

**F.1.1.45 Shipping containers\*:** especially designed, pre-moulded skids or containers, that are used to hold and contain the finished cast parts during shipment. These may be provided by the customer or by the foundry.

**F.1.1.46 Trim:** unwanted material that is cut away from the desired portions of the raw casting.

EXAMPLE Trim includes gates, risers, runners, and ash on the raw casting.

**F.1.1.47 Assemble patterns and rigging onto plate (A1214):** mount the patterns and the rigging components onto the pattern plate.

**F.1.1.48 Build patterns (A1212):** produce the patterns required to make moulds for sand casting.

**F.1.1.49 Build pattern and rigging assembly (A121):** make a master of the pattern, if it is required, construct the desired patterns, fabricate the needed rigging components, and mount the patterns and the rigging components onto the pattern plate.

**F.1.1.50 Build rigging (A1213):** construct the rigging components required for the pattern and rigging assembly.

**F.1.1.51 Build sand mould (A12):** the sequence of processes used to produce the completed mould in sand casting. This includes building the pattern and rigging assembly, making the sand mould, placing cores, filters and inserts into the sand mould, and inspecting and preparing the mould for casting.

NOTE The activity "Make sand mould", described later, is more limited in meaning, and is concerned only with the production of the empty cope and drag halves of the mould, by forming sand against the pattern and rigging assembly.

**F.1.1.52 Cores and inserts design:** the complete geometric description of the cores and inserts, their location in the sand mould, the specified core or insert materials, and the process specified for their production.

NOTE A description of any masters used to create core moulds is also included.

**F.1.1.53 Data required for simulation of sand casting:** a complete description of the geometry of the sand casting, rigging components and attributes which characterize the materials in contact with the metal at any particular surface of the casting.

NOTE Attributes which characterize the materials would include a description of sand, chills, sleeves, filters and cores in the sand mould.

**F.1.1.54 Description of master of pattern:** information needed to characterize the design of the master which will be constructed during pattern manufacture. This includes the measured dimensions, the material to be used for its manufacture, and any identifications assigned to the master.

NOTE The master is a scaled form, which is used to produce a sand casting pattern of the desired size and shape, often by machining while using the master as a template.

**F.1.1.55 Description of masters of cores:** information needed to characterize the design of masters. This includes the measured dimensions, the material to be used for their manufacture, and any identifications assigned to the master.

NOTE The master is a scaled form, which is used to produce a core box of the desired size and shape, often by machining while using the master as a template.

**F.1.1.56 Design cores and inserts for sand mould (A1112):** specify the geometry of the cores and inserts, their location in the sand mould, and the materials to be used in their production.

**F.1.1.57 Design pattern for sand mould (A1111):** specify the complete geometric description of the pattern, the location of the pattern or patterns on the pattern plate, and the process to be used for the production of the patterns.

NOTE The design pattern process includes describing the geometry of the master, selecting the parting plane or parting planes, deciding whether a single pattern or multiple pattern casting is desired, describing the geometry of the pattern or patterns on each side of the plate, and assigning the location of the patterns on the pattern plate.

**F.1.1.58 Design rigging and secondary tooling for sand mould (A1113):** specify the geometry of the components of the rigging, their location with respect to the pattern on the pressure or match plate, and the materials to be used for their construction. Establish the design of secondary tooling required for finishing of the raw sand casting, the geometry of each tooling item, and the materials to be used for their construction.

NOTE The secondary tooling devices are used in finishing and inspection, and include fixtures for trim or straightening presses, machining fixtures, finishing fixtures, gauging fixtures, heat treating fixtures, core removal fixtures, painting fixtures, plating fixtures, shipping containers and leak testing fixtures.

**F.1.1.59 Design sand mould (A111):** specify the complete design of the sand mould, that consist of the design of the pattern or patterns, the rigging, cores and inserts. The design of the mould flask and the pattern plate is also included here.

**F.1.1.60 Design sand mould and plan process (A11):** specify the sand mould design, simulate the sand casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE The sample castings are analyzed and tested in detail to provide quality assurance data, that may be submitted to the customer in order to qualify the foundry's mould design and possibly their process plan for subsequent large scale production of the casting part.

**F.1.1.61 Filled sand mould\*:** the sand mould and its contents immediately after molten metal has been poured into it, and before the metal has solidified and cooled to the point where it can be shaken out of the mould.

**F.1.1.62 Inspect and prepare mould (A125):** the sand mould is examined for any defects, such as loose sand or cracks, and for dimensional tolerances.

NOTE In processes which use bonded sand or plaster, the mould is allowed to harden or cure. If mould coatings are to be used, they are applied at this time. Cores, inserts and filters are set into the mould, and their placement is checked. The mould is given a final inspection and the mould is closed. Weights are placed on top of the mould, and brackets may also be placed at the top or sides of the mould. The mould is now ready to receive liquid metal.

**F.1.1.63 Make cores (A123):** manufacture cores for sand casting.

**F.1.1.64 Make masters (A1211):** build core or pattern masters.

**F.1.1.65 Make sand mould (A122)\*:** manufacture the mould cavity used in sand casting.

NOTE The mould consists of a hollow cavity, which is constructed by forming moulding sand against both sides of the pattern and rigging assembly or pressure plate, removing the pattern, and the reassembling the top and bottom halves of the sand mould. This activity also includes the placement of any riser sleeves, chills or filters into the mould.

**F.1.1.66 Master of cores:** a scaled form used to produce a core box, often by machining while using the master as a template.

**F.1.1.67 Master of pattern:** a scaled form of the pattern used to produce the pattern, by casting impressions of the master, or by machining while using the master as a template.

**F.1.1.68 Materials for masters:** the materials used to produce core or pattern masters.

**F.1.1.69 Materials for pattern, rigging, mould, inserts and cores for sand casting:** the materials used to construct the pattern, the rigging, the mould, inserts and cores in sand casting.

**F.1.1.70 Melt, pour, cool and shakeout (A13)\*:** prepare liquid metal alloy, pour it into the sand mould, allow time for the sand casting to freeze and cool, and remove or shake out the raw sand casting from the sand mould.

**F.1.1.71 Moulding machines:** the equipment used to make sand moulds.

**F.1.1.72 Patterns:** shaped component used to produce moulds in sand casting.

NOTE There are three types of pattern or pattern plate assemblies. The first type is called a pressure plate. The pressure plate is usually made of one piece, by casting a plate together with impressions made from the pattern master. The pressure plate can also be made by machining. The second type of pattern is called a match plate. In this case master pieces, or copies of the master, are bolted or screwed onto the pattern plate. The third type of pattern is a loose pattern, where the pattern pieces are placed on a board, but not fixed permanently thereon. Loose patterns are normally employed when only a few castings are needed.

**F.1.1.73 Pattern and rigging assembly:** the completed assembly, used to make sand moulds, in which the pattern or patterns and the rigging have been mounted or placed on the pattern plate.

**F.1.1.74 Pattern and rigging materials:** the materials used to construct the pattern or patterns and the rigging for sand casting.

**F.1.1.75 Pattern design:** information that characterizes the design of the sand casting patterns.

NOTE The information includes a complete geometric description of the master used to make the pattern, the location of patterns on the pattern plate, the material to be used in constructing the patterns, the size of the mould {cope and drag heights} and the pattern plate size.

**F.1.1.76 Pattern materials:** the materials used to construct the sand casting patterns.

NOTE Wood, plastic and aluminum are commonly used materials.

**F.1.1.77 Place cores and inserts into mould (A124):** arrange cores and inserts inside the sand mould.

**F.1.1.78 Pour and cool sand casting (A132)\*:** pump or ladle liquid metal into the sand mould, and allow time for the casting to solidify and cool.

**F.1.1.79 Pouring parameters\*:** quality control process data that records the pouring process information for a casting.

NOTE The parameters include the heat number, the temperature of the metal as it is poured into the mould, and the time to fill a mould during pouring.

**F.1.1.80 Preliminary sand mould assembly:** the sand mould assembly before final inspection and any curing of the mould.

NOTE The preliminary sand mould assembly consists of the assembled cope and drag sections of the mould, inserted cores, inserts and chills, and any other specified rigging components.

**F.1.1.81 Prepare melt for sand casting (A131)\*:** charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the sand mould.

**F.1.1.82 Prepared sand mould\*:** an assembled sand mould ready to receive liquid metal.

NOTE The prepared sand mould contains the needed cores, chills, inserts and rigging components, has been inspected; and in case of bonded sand or plaster mould casting, it has also been cured.

**F.1.1.83 Raw sand casting:** the sand casting after it has been removed from the mould, and before finishing or inspection operations.

**F.1.1.84 Recycled sand (A134)\*:** sand that is recovered during shakeout of the casting, and which is recycled back into the foundry sand system.

**F.1.1.85 Rigging:** parts of the sand mould, aside from the pattern or patterns.

NOTE The rigging thus includes items used to convey the molten metal in the mould such as the pouring cup, sprue, basins, gates, runners, risers and vents as well as other miscellaneous mould components such as chaplets or nails, internal and external chills, test bars, insulating sleeves, in-mould filters, and in-mould additions of flux, grain refiner, or inoculant.

**F.1.1.86 Rigging design:** the complete geometric description of the rigging components, including their location on the pressure plate or in the mould, and a description of the materials to be used in construction of the rigging components.

**F.1.1.87 Rigging materials:** materials used to construct the rigging components, which are mounted or placed on the pattern plate in sand casting.

**F.1.1.88 Sample and plan sand casting process (A113)\*:** produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

**F.1.1.89 Sand:** includes all materials used to make moulds from a non-consumable pattern and rigging assembly; including the usual "green" sands; which are usually a mixture of silica or zircon sand, clay and water; bonded sands, and plaster.

NOTE When a consumable wax pattern is employed to produce a plaster mould, the casting process is categorized under investment casting, not sand casting.

**F.1.1.90 Sand and materials for mould\*:** the sand mixture and other materials used to construct the mould in sand casting.

**F.1.1.91 Sand cast a part (A1):** the sequence of activities and materials required to manufacture a product by sand casting.

**F.1.1.92 Sand casting process data:** information used to characterize the design and manufacture of the pattern, rigging and moulding.

NOTE the melting and pouring processes to produce a sand casting are out-of-scope.

**F.1.1.93 Sand mould\*:** a hollow cavity constructed by forming the moulding sand against both sides of the pattern and pattern plate.

**F.1.1.94 Sand mould design:** the information that characterizes the design of the sand casting pattern, the cores, inserts and the rigging; including information regarding the location of the components with respect to the parting plane, and the location of the components on the pattern plate.

**F.1.1.95 Sand mould inspection and preparation results:** information regarding the results of the final inspection and preparation of the sand mould.

**F.1.1.96 Sand mould quality control data:** quality control process data, that describe the processing of the sand mould, the cores, and inspection of the sand mould.

**F.1.1.97 Sand mould processing data\*:** quality control process data that characterizes the properties of the sand used in construction of the mould.

NOTE This processing data includes the measured composition of the sand mix, the results of tests to determine sand strength, compactability, permeability, and the measured grain size of the sand. Process Information about the sand mulling and sand moulding operations is also included in sand mould processing data.

**F.1.1.98 Sand mould quality control data\*:** quality control process data that characterizes the properties of the sand used in construction of the mould, the core processing data, and the sand mould inspection and preparation results.

**F.1.1.99 Sand recycle equipment\*:** equipment used to recover used sand in the foundry.

NOTE Sand heat exchangers and screening equipment fall into this category.

**F.1.1.100 Shake out sand casting (A133):** after cooling, the filled sand mould is placed into a machine, which shakes the casting in order to separate it from the sand in the mould and cores.

**F.1.1.101 Shakeout machine:** equipment used to remove the sand from raw casting or knock out a casting.

**F.1.1.102 Shakeout machine and sand recycle equipment\*:** equipment used to shake out a casting, and to recycle the sand recovered in the shakeout process.

**F.1.1.103 Simulate sand casting (A112):** model the sand casting process, by using a computer-based simulation of the mould filling and metal freezing processes which occur during sand casting.

**F.1.1.104 Simulation results for sand casting:** predictions of casting quality made by using a computer-based model of the mould filling and metal freezing processes that occur during sand casting.

NOTE Typical sand casting simulation results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format.

**F.1.1.105 Used sand\*:** the sand resulting from the shakeout process.

**F.1.1.106 Waste sand\*:** sand that is not of the proper size, or is otherwise unsuitable, to be used for the moulding material in sand casting.

**F.1.1.107 Assemble die components (A223):** assemble the finished die mould assembly from all individual die components assembled together.

**F.1.1.108 Build die (A22):** produce and assemble the finished die mould assembly in the die casting machine.

**F.1.1.109 Build die components (A221):** manufacture the die components by casting, machining or hipping, to bring them to the final desired shape, dimensions and surface finish.

**F.1.1.110 Cool and extract (A234):** allow the molten metal in the die to solidify and cool, and then extract or eject the solidified casting from the die.

**F.1.1.111 Data required for simulation of die casting:** a complete description of the geometry of the die casting and the die mould, proposed location of any cooling channels, attributes that characterize the materials in contact with the metal at any particular surface of the casting.

NOTE Attributes which characterize the materials would include a description of chills, sleeves, filters and mould coatings in the die.

**F.1.1.112 Design die (A211):** consists of designing the layout of die sub assemblies in the die, and designing all sub assemblies, components, and the die rigging.

**F.1.1.113 Design die and plan die casting process (A21):** specify the die design, simulate the die casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE The sample castings are analyzed and tested in detail to provide quality assurance data, which may be submitted to the customer in order to qualify the foundry's die design and their process plan for subsequent large scale production of the casting part.

**F.1.1.114 Design die layout (A2111):** establish the overall design and layout of the complete die mould assembly.

NOTE This design includes a list of all die sub-assemblies needed for the die mould, layout and assembly drawings, and material or stock lists of needed components and sub assemblies.



**F.1.1.115 Design die rigging and secondary tooling (A2113):** specify the die rigging design and the secondary tooling design.

**F.1.1.116 Design die sub-assemblies (A2112):** establish the design of die mould sub-assemblies required for the die mould.

NOTE Sub-assemblies would consist of items such as: fixed cores, slides, consumable cores, inserts, core pin layouts, ejector pin and plate assemblies, coolant layouts, hydraulic layouts, electrical layouts and safety switch layouts.

**F.1.1.117 Die assembly:** consists of all individual die components assembled together.

**F.1.1.118 Die assembly data:** production information related to the act of constructing the die assembly.

NOTE Information included in die assembly data is the name and identification number of the assembly, personnel responsible for assembly, and inspection results.

**F.1.1.119 Die assembly design:** completed design of the entire die mould assembly, that includes the die layout, the die sub-assemblies design and the die rigging design.

**F.1.1.120 Die cast a part (A2):** sequence of activities and materials required to manufacture a product by die casting.

**F.1.1.121 Die casting machine:** equipment that houses the die assembly and produces die castings.

**F.1.1.122 Die casting process data:** information used to characterize the design and manufacture of the die, and the processes employed to produce a die casting.

**F.1.1.123 Die components:** the individual cast or machined components, before any heat treatment, that together are used to produce the final die assembly in the die casting machine.

NOTE Die components include items such as mould sections, cores or slides, the shot sleeve and ejector pins.

**F.1.1.124 Die layout:** specifies the overall layout of the die assembly, and how the part to be cast is placed inside the die mould.

**F.1.1.125 Die or mould temperature control system\*:** the machinery and controls used to cool or heat the mould in die casting, thereby bringing it to the desired temperature for pouring the casting.

**F.1.1.126 Die preparation process data\*:** describes the processes and procedures used to prepare the die before casting begins, including data on the measured temperature of the mould and on the application of mould coatings.

**F.1.1.127 Die property data:** relevant information regarding the construction and assembly of the die mould used in die casting.

NOTE Die property data include data on the finished die components, the heat treatment of die components, the construction of the die assembly, and the process of mounting the die assembly on the die casting machine.

**F.1.1.128 Die rigging design:** the design and geometric description of the die rigging, that includes the size and material of the shot sleeve, location of the sprue hole, and the size and location of the runners, gates, overflows, risers and vents.

NOTE In a sense, the die rigging may be considered to be a special sub-assembly.

**F.1.1.129 Die sub-assemblies design:** a complete description of the geometry or other design of die sub-assemblies, together with a specification of the materials to be used in construction of the sub-assemblies.

NOTE Typical sub-assemblies are: fixed cores, slides, consumable cores, inserts, core pin layouts, ejector pin and plate assemblies, coolant layouts, hydraulic layouts, electrical layouts, and safety switch layouts.

**F.1.1.130 Die temperature control system\*:** the machinery and controls used to cool or heat the mould or die in die casting; thereby bringing it to the desired temperature for pouring the casting.

**F.1.1.131 Filled die mould\*:** the die mould and its contents immediately after molten metal has been shot into it and before the metal has solidified and cooled to the point where it can be extracted.

**F.1.1.132 Finished die components data:** information related to the production of die components used to construct the die assembly.

NOTE Finished die components data include the name of the component, the supplier or manufacturer, and relevant information about the materials used to manufacture the component.

**F.1.1.133 Heat treat data:** quality control process data related to the heat treatment of individual die mould components.

NOTE Examples of heat treat data are the times and temperatures employed in the heat treatment process, and the results of hardness or toughness measurements.

**F.1.1.134 Heat treat die components (A222)\*:** subject the specified die components to a sequence of high temperature treatments in order to produce the desired mechanical properties in the component.

NOTE Heat treatment of tool steel die components for high pressure die casting would normally consist of a high temperature austenization treatment, followed by quenching and possibly tempering or stress relieving. Die components for low pressure die casting generally do not receive a heat treatment.

**F.1.1.135 Heat treated die components:** die components after they have received the specified heat treatment.

**F.1.1.136 Materials for cores and die coatings:** the materials used to produce cores, and to coat the die or permanent mould in die casting.

**F.1.1.137 Materials for cores, die coatings, and die components:** the materials used to produce accessory components for the die in die casting.

**F.1.1.138 Materials for die components:** the materials used to construct individual die components in die casting.

**F.1.1.139 Melt, pour, cool and extract (A23)\*:** prepare liquid metal alloy, pour it into the die mould, allow time for the die casting to freeze and cool, and then extract the casting from the die mould.

**F.1.1.140 Mould temperature control system\*:** used to bring the die casting mould to the desired temperature prior to pouring liquid metal into the mould.

**F.1.1.141 Mount die assembly in machine (A224):** fix into position the completed die assembly into the die casting machine.

NOTE At times the die assembly occurs directly on the die casting machine. In this case the die assembly and die mounting processes are combined.

**F.1.1.142 Mounted die assembly:** the completed die assembly, after it has been mounted on the die casting machine.

**F.1.1.143 Mounted die assembly data:** relevant manufacturing information regarding the location of the mounted die assembly on the die casting machine, that includes die assembly identification number and the die casting machine number.

**F.1.1.144 Pour shot (A233)\*:** inject, pump or ladle molten metal into the die mould.

NOTE In the case of high pressure die casting this activity includes the use of a ram to force molten metal into the mould.

**F.1.1.145 Prepare die (A232):** apply lubricant or die coatings and heat the die to the desired temperature.

**F.1.1.146 Prepare melt for die casting (A231)\*:** charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the die mould.

**F.1.1.147 Prepared die:** the die that has been heated to the proper temperature and coated with suitable mould coatings and made ready to receive liquid metal for the first time.

**F.1.1.148 Process data on cycle time\*:** the information that describes the time between shots into the die.

**F.1.1.149 Raw die casting:** the die casting after it is removed from the mould, and before any trimming, grinding, finishing, machining, heat treating or inspection.

**F.1.1.150 Sample and plan die casting process (A213):** produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

**F.1.1.151 Scrapped castings:** castings that do not to meet the customer's specifications and requirements, and that cannot be repaired.

**F.1.1.152 Shot data\*:** quality control process data recorded during the introduction of molten metal into the die casting mould.

NOTE In high pressure die casting, a detailed profile of the ram velocity is usually recorded together with the pressure at end of fill and the intensification pressure. In gravity pour die casting, the fill time is measured. In tilt-pour permanent mould casting, the tilt rate and tilt profile is recorded.

**F.1.1.153 Shot process data\*:** quality control information describing the shot data, the die process data, and process data on the cycle time.

**F.1.1.154 Shot process design parameters\*:** the detailed design characterization of how liquid metal shall be introduced into the die mould, and the desired metal pouring temperature.

NOTE In high pressure die casting the shot process parameters specify the ram speed profile and the ram intensification pressure. In low pressure or gravity pour die casting, the shot process parameters specify the fill time. In tilt-pour die casting, the shot process parameters specify the tilt rate of the mould.

**F.1.1.155 Simulate die casting (A212):** model the die casting process, by using a computer-based simulation of the mould filling and metal freezing processes which occur during die casting.

**F.1.1.156 Simulation results for die casting:** predictions of casting quality made by using a computer-based model of the mould filling and metal freezing processes which occur during die casting.

NOTE Typical die casting simulation results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format. The die casting simulation results would include any predictions of die distortion or cracking, and estimates of the cooling needed at various points in the die.

**F.1.1.157 Build investment pattern (A321):** produce the consumable pattern, that is covered with a ceramic mould in the investment casting processes.

NOTE The pattern is normally built by injecting wax or foam into the pattern die.

**F.1.1.158 Build investment pattern assembly (A322):** construct the investment pattern assembly, that consists of mounting the pattern or patterns onto the rigging assembly.

NOTE The investment rigging assembly includes the pouring cup, down sprue, gates, runners and risers.

**F.1.1.159 Build plaster mould (A331):** produce a plaster mould, by pouring plaster around the investment pattern assembly and allowing the plaster to harden and cure.

**F.1.1.160 Build wax or foam pattern (A32):** produce the consumable wax or foam pattern, that is used to make ceramic moulds in the investment casting processes.

**F.1.1.161 Compaction process data for investment casting:** quality control process data that records the information related to the sand compaction process, in lost foam investment casting.

NOTE This should not be confused with the compactability test commonly used in sand casting, to characterize the quality of the moulding sand.

**F.1.1.162 Cores and investment pattern die:** cores are components added to the wax pattern before investment to produce hollow sections in the casting. The pattern die is injected with wax or foam, to produce consumable patterns.

**F.1.1.163 Cure shell or plaster mould (A335)\*:** subject the de-waxed investment mould to a high temperature heat treatment to improve the strength of the ceramic shell or plaster mould and remove any remaining wax or moisture present in the mould.

**F.1.1.164 Curing process data\*:** quality control process data recorded about the mould curing process in investment casting.

NOTE The curing process data includes the times and temperatures used in the curing furnace.

**F.1.1.165 Data required for simulation of investment casting:** a complete description of the geometry of the complete investment casting, or pattern assembly. Also included would be a description of the mould, and the temperature of the mould at pouring.

**F.1.1.166 Design investment cores and secondary tooling (A3113):** specify the investment core design and the secondary tooling design.

**F.1.1.167 Design investment mould (A311):** specify the complete geometric description of the investment pattern, the design of components to be used to build the investment pattern assembly rigging, the location and orientation of patterns on the pattern assembly, and the design of any investment cores required. The processes to be used for the production of the components is also specified at this time. Any secondary tooling required to process the raw investment casting is also established.

**F.1.1.168 Design investment pattern and pattern die (A3111):** specify the complete geometric description of the investment pattern die, and any process parameters to be used for the production of patterns from the die. The geometry of the consumable pattern is also established at this time. This design activity also includes the development of any masters that may be used to produce investment pattern dies.

NOTE Wax or foam is injected into the pattern die to produce investment patterns.

**F.1.1.169 Design investment pattern assembly and plan process (A31):** specify the investment pattern assembly design, simulate the investment casting process, produce a small production run of sample castings, and establish the process plan to be used in large scale production of castings.

NOTE The sample castings are analyzed and tested in detail to provide quality assurance data, which may be submitted to the customer in order to qualify the foundry's mould design and possibly their process plan for subsequent large scale production of the casting part.

**F.1.1.170 Design pattern assembly (A3112):** specify the complete geometric description of the components of the investment pattern assembly, and processes to be used for their production.

NOTE The investment pattern assembly includes the pouring cup, down sprue, gates, runners and risers. Also specified in this design activity is the placement of the pattern or patterns on the rigging assembly.

**F.1.1.171 Dewax (A333):** remove the wax pattern from the shell or plaster mould, by heating the green plaster mould or wax shell mould in a furnace or autoclave.

**F.1.1.172 Dewaxing process data:** quality control process data about the process used to remove the wax from a green plaster mould, or a wax shell mould, in investment casting.

NOTE The dewaxing process data would include the steam pressure and time to reach pressure in an autoclave, or furnace temperature and treatment time in the ash firing process.

**F.1.1.173 Die or mould temperature control system\*:** the machinery and controls used to cool or heat the ceramic mould in investment casting, thereby bringing it to the desired temperature for pouring the casting.

**F.1.1.174 Filled investment mould\*:** The investment mould and its contents immediately after molten metal has been poured into it, and before the metal has solidified and cooled to the point where it can be knocked out of the mould.

**F.1.1.175 Foam investment:** the foam pattern assembly, after it has been invested with a ceramic shell, but before it is packed in sand.

NOTE Foam investment is a green shell mould made from a foam pattern assembly.

**F.1.1.176 Green plaster mould:** the investment pattern assembly after it has been invested with plaster, but before it has been dewaxed or subjected to a high temperature curing process.

NOTE Wax is commonly used to construct the pattern assembly for plaster mould investment casting.

**F.1.1.177 Green shell mould:** the wax or foam pattern assembly, immediately after it has been invested with a ceramic shell.

**F.1.1.178 Heat shell or plaster mould (A342)\*:** place the plaster mould or shell mould in a furnace, in order to bring it to the desired temperature to pour an investment casting.

**F.1.1.179 Heated shell or plaster mould\*:** the shell or plaster mould in investment casting, after it has been heated to the desired temperature for casting, and before it has received any liquid metal.

**F.1.1.180 Inspect investment pattern assembly (A323):** examine the final pattern assembly or pattern tree for dimensional tolerance and completeness before investment.

**F.1.1.181 Invest pattern (A33):** coat or invest the foam or wax pattern assembly with a ceramic shell or a plaster mould, dewax and cure the mould and; in the case of lost foam casting, compact sand around the mould.

**F.1.1.182 Invest with ceramic shell (A332):** coat or invest a foam or wax investment pattern assembly with a ceramic shell mould coating.

NOTE The shell mould coating usually consists of 6-8 thin layers of ceramic backed with a stucco coating.

**F.1.1.183 Investment cast a part (A3):** the sequence of activities and materials required to manufacture a product by investment casting.

**F.1.1.184 Investment casting mould design:** the design of the complete investment mould assembly, which consists of the investment pattern, investment pattern dies used to produce patterns, the investment pattern assembly, and investment cores.

**F.1.1.185 Investment casting process data:** information used to characterize the design and manufacture of investment patterns and investment pattern assemblies.

NOTE the processes employed to produce an investment casting are out-of-scope.

**F.1.1.186 Investment cores design:** a complete description of the geometry of cores used in investment casting, the materials used in their construction, and specifications regarding the procedures used to manufacture the cores.

**F.1.1.187 Investment data:** quality control process data recorded about the process of investment casting.

NOTE This information includes data on the manufacturing of plaster and ceramic shell investment moulds, and on the dewaxing and curing of moulds. Also included is data on the sand compaction process in lost foam investment casting.

**F.1.1.188 Investment masters design:** a complete geometric description of any masters used in investment casting. Masters might be used to produce pattern dies, or cores.

**F.1.1.189 Investment materials:** the materials used to invest the pattern and rigging assembly.

NOTE The materials most commonly used to build ceramic investment moulds are colloidal silica gel-, ethyl silicate-, and gypsum-bonded ceramics.

**F.1.1.190 Investment pattern:** a consumable pattern, constructed of wax, plastic or foam, that has the same shape as the final desired cast product, but with slightly different dimensions to allow for thermal contraction of the casting after solidification. The pattern is invested with ceramic to produce part of the mould used in investment casting.

**F.1.1.191 Investment pattern assembly:** the completed wax or foam pattern assembly before investment. This includes the investment pattern or patterns and the investment rigging assembly. The investment rigging assembly includes the pouring cup, down sprue, gates, runners and risers.

NOTE The investment pattern assembly is often called a pattern tree and is constructed from wax, plastic or foam.

**F.1.1.192 Investment pattern assembly data:** the quality control process data related to the production of the finished investment casting pattern assembly.

NOTE This information includes data which characterizes the processes used to produce patterns, by injecting wax or foam into pattern dies, and the results of inspecting the final investment pattern assembly.

**F.1.1.193 Investment pattern assembly design:** the complete geometric description of the investment rigging components; including the pour cup, down sprue, gates, runners and risers; and the location and orientation of rigging components in the final pattern assembly.

NOTE The location of patterns in the final pattern assembly, or pattern tree, is also specified at this time. Also included are the processes specified for production of rigging components, and the desired materials to be used in the investment pattern assembly.

**F.1.1.194 Investment pattern assembly inspection results:** quality control process data related to the inspection of the preliminary investment pattern assembly.

**F.1.1.195 Investment pattern design:** the complete geometric description of the investment pattern, and the process specified for pattern production.

**F.1.1.196 Investment pattern die design:** the complete geometric description of dies used to create the wax or foam, and the materials to be used in their construction.

**F.1.1.197 Investment pattern process data\*:** quality control process data related to the production of investment casting patterns.

NOTE The information includes die injection pressures used, cycle times, and the temperature of the injected wax or foam. In production of foam patterns this includes the vacuum applied during the pre-expansion of the beads, and the pattern moulding parameters such as steam temperature, time of steam application, die cycle time and pattern aging time.

**F.1.1.198 Knockout investment casting (A344):** place the cooled investment mould into a machine, that shakes or vibrates the mould, in order to break up the mould and separate it from the casting.

NOTE Any washing procedures would also be included in this activity. F.1.1.205 Lost foam mould: the mould produced for lost foam casting by investing a foam pattern assembly, and compacting sand around the foam investment.

**F.1.1.199 Materials for wax or foam pattern and investment:** The materials used to construct the consumable pattern, and the ceramic investment or mould materials in investment casting.

NOTE Wax and foam and foam are commonly used to produce investment patterns, although other materials may be used in special applications.

**F.1.1.200 Melt, pour, cool and knockout (A34)\*:** prepare liquid metal alloy, pour the metal into the investment casting mould, allow time for the casting to freeze and cool, and remove or knock out the casting from the ceramic mould.

**F.1.1.201 Plaster investment process data\*:** quality control process data that records the processes used to manufacture the plaster mould, in investment casting.

NOTE This information would contain the composition of the plaster, vibration and vacuum applied, if any, during investment and the time allowed for plaster to harden.

**F.1.1.202 Pour and cool investment casting (A343)\*:** pump or ladle liquid metal into the investment mould and allow time for the casting to solidify and cool.



**F.1.1.203 Pouring parameters\*:** quality control process data that records the pouring process information for a casting.

NOTE This data includes the heat number, the temperature of the metal as it is poured into the mould, and the time to fill a mould during pouring. In investment casting it also includes the mould temperature prior to pouring.

**F.1.1.204 Preliminary investment pattern assembly:** The investment pattern assembly before inspection.

**F.1.1.205 Prepare melt for investment casting (A341)\*:** charge materials are melted, refined, held at desired temperature and brought to the desired alloy composition in preparation for pouring into the investment mould.

**F.1.1.206 Raw investment casting:** the investment casting after it is removed from the mould, and before any finishing or inspection.

**F.1.1.207 Recycled wax\*:** wax that is recovered during the dewaxing process, and that is reused in pattern or rigging assembly components.

**F.1.1.208 Sample and plan investment casting process (A313):** Produce a small sample lot of castings, test and inspect the castings, and use the test results to establish an effective process plan, which can be used to produce quality castings in large quantities.

**F.1.1.209 Sand compaction for lost foam compaction (A334):** the process of placing loose sand around the foam investment, and vibrating the sand to compact it around the lost foam mould.

**F.1.1.210 Sand for lost foam compaction\*:** loose sand, that is compacted around the lost foam mould in order to hold and support the mould during the pouring process.

**F.1.1.211 Shakeout machine\*:** equipment used to shake out or knock out a casting.

**F.1.1.212 Shell coating process data\*:** information recorded about the manufacture of the shell coating in investment casting.

NOTE This information includes the composition of material used in each layer of the shell, any stucco materials applied, and drying times and drying conditions between layers.

**F.1.1.213 Shell mould or plaster mould:** two types of moulds used in investment casting, that are dewaxed and cured before casting.

NOTE The shell mould is constructed by applying a number of thin ceramic coatings to the investment pattern assembly. The plaster mould is constructed by immersing the wax investment pattern assembly into a container filled with freshly mixed plaster, and allowing the plaster to harden. As used here plaster includes both gypsum bonded ceramics and other comparable ceramic slurries.

**F.1.1.214 Simulate investment casting (A312):** model the investment casting process, by using a computer-based simulation of the mould filling and metal freezing processes that occur during investment casting.

**F.1.1.215 Simulation results for investment casting:** predictions of casting quality made by using a computer-based model of the mould filling and metal freezing processes that occur during investment casting.

NOTE Typical investment casting results would include predictions related to casting quality, such as the possible location of hot spots or shrinkage cavities, and the distribution of porosity; and may be in a text or pictorial format.

**F.1.1.216 Uncured plaster or shell mould:** the plaster or ceramic shell mould, after dewaxing and before the high temperature mould curing process.

**F.1.1.217 Wax or foam:** the wax or foam materials used to construct the pattern assembly in investment casting.

**F.1.1.218 Wax shell mould:** a mould in investment casting that has been made by constructing a wax pattern assembly, and investing the assembly with a ceramic shell coating.

NOTE Wax shell mould is a green shell mould made from a wax pattern assembly.

## **F.2 Application activity model diagrams**

The application activity model is given in Figures F.2 through F.14. Activities and data flows that are out of scope are marked with asterisks.

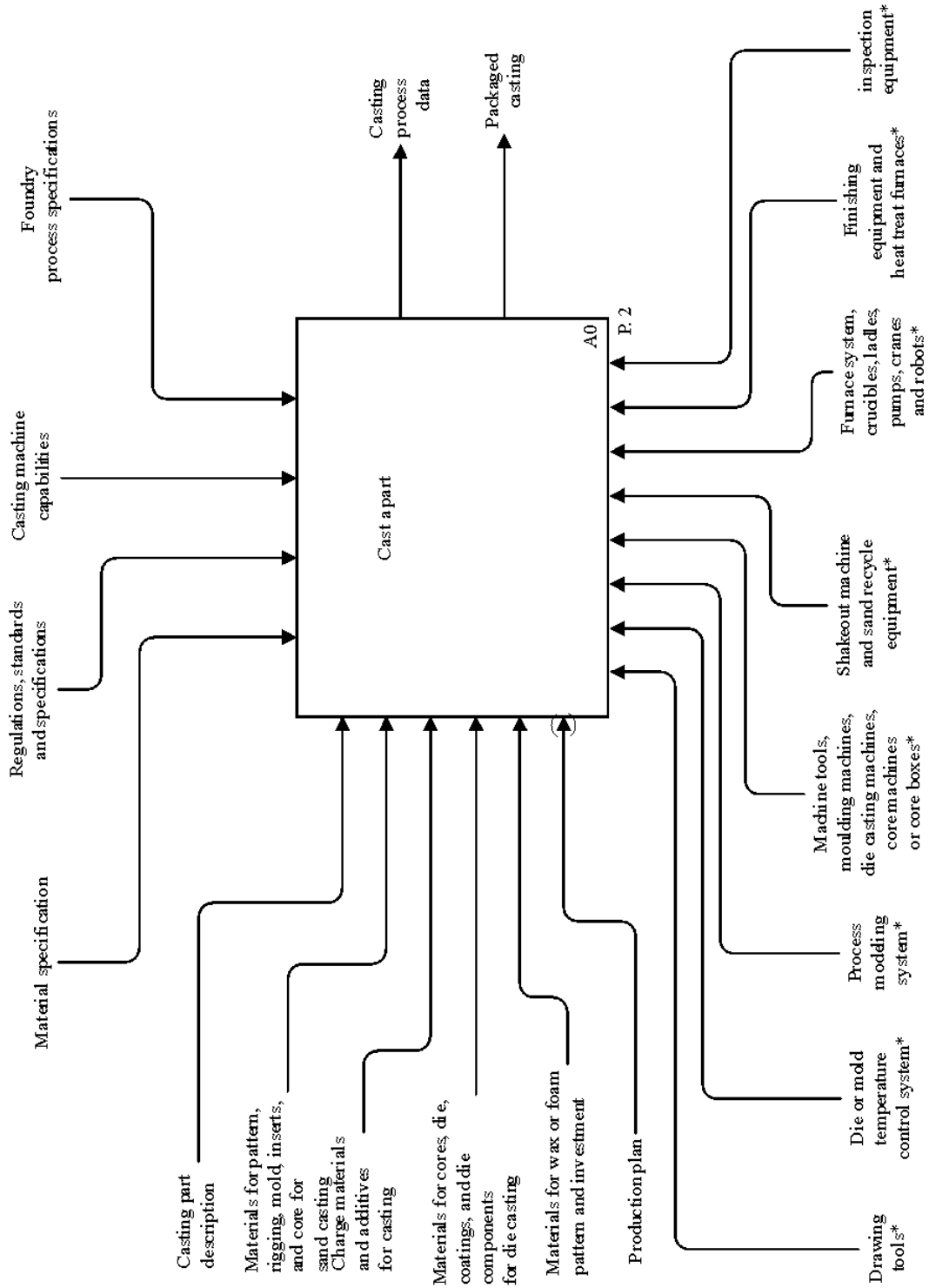


Figure F.2 - A-0 Exchange of design and manufacturing product information for cast parts

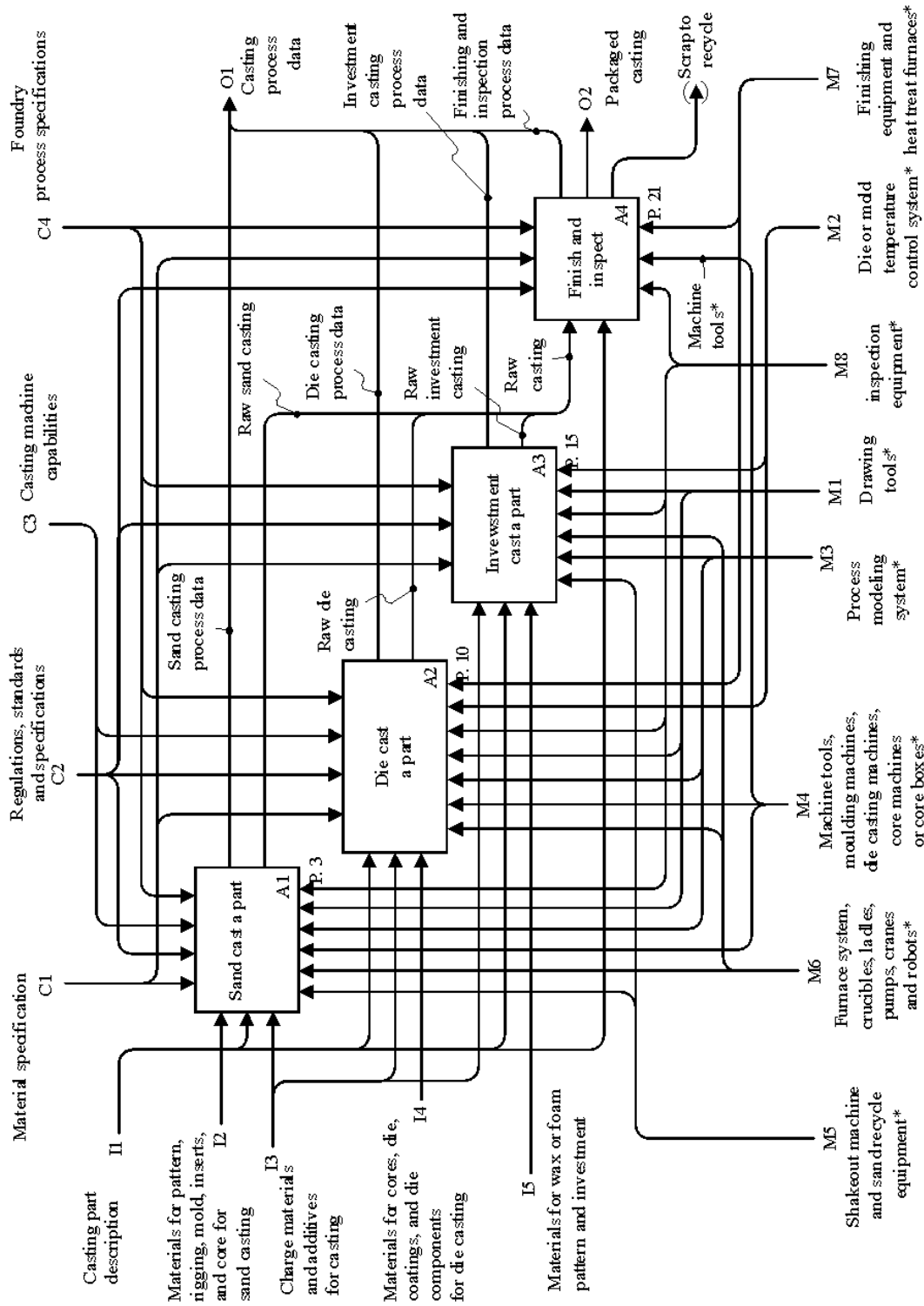


Figure F.3 - A0 Cast a part

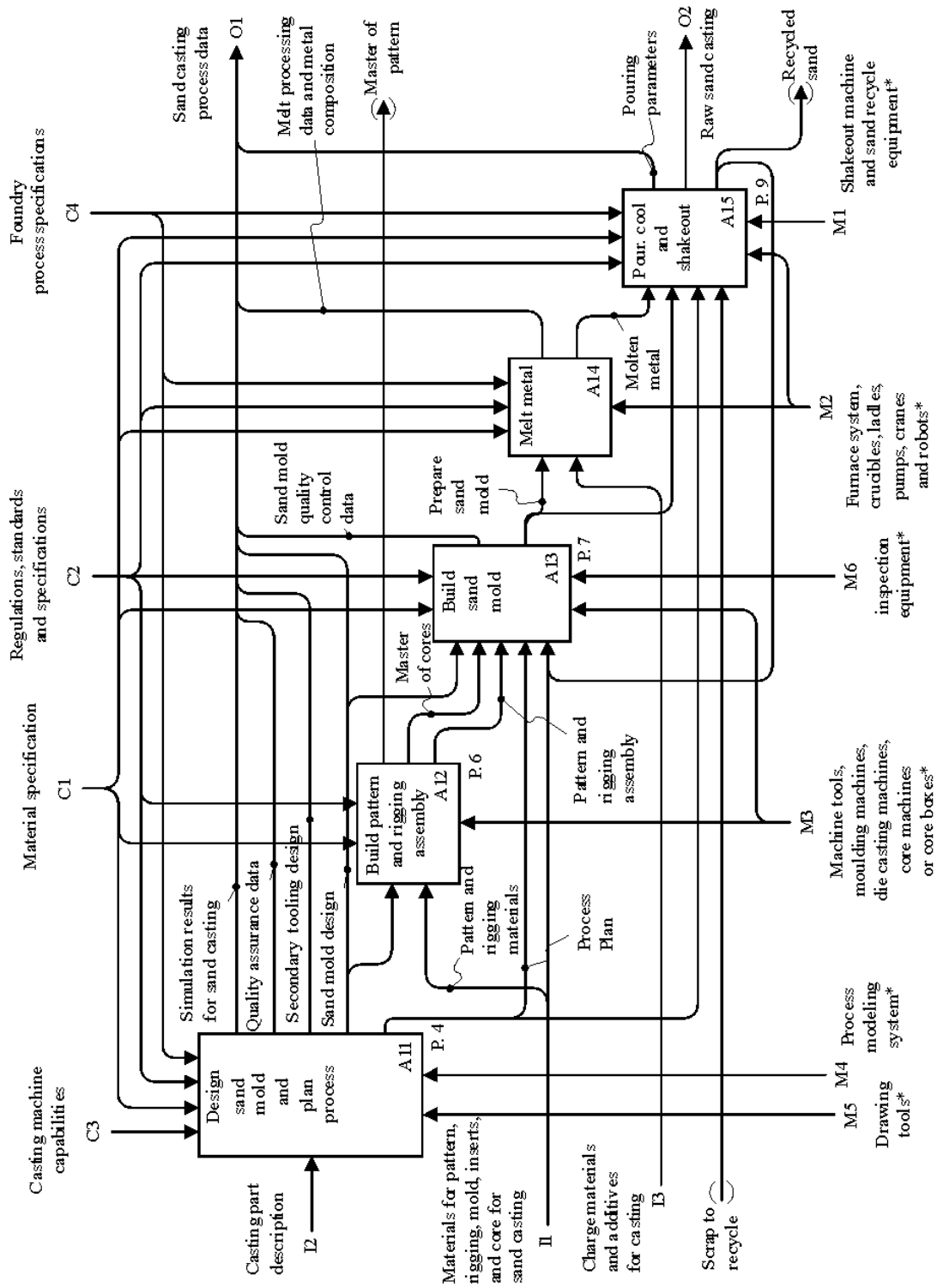


Figure F.4 - A1 Sand cast a part

F

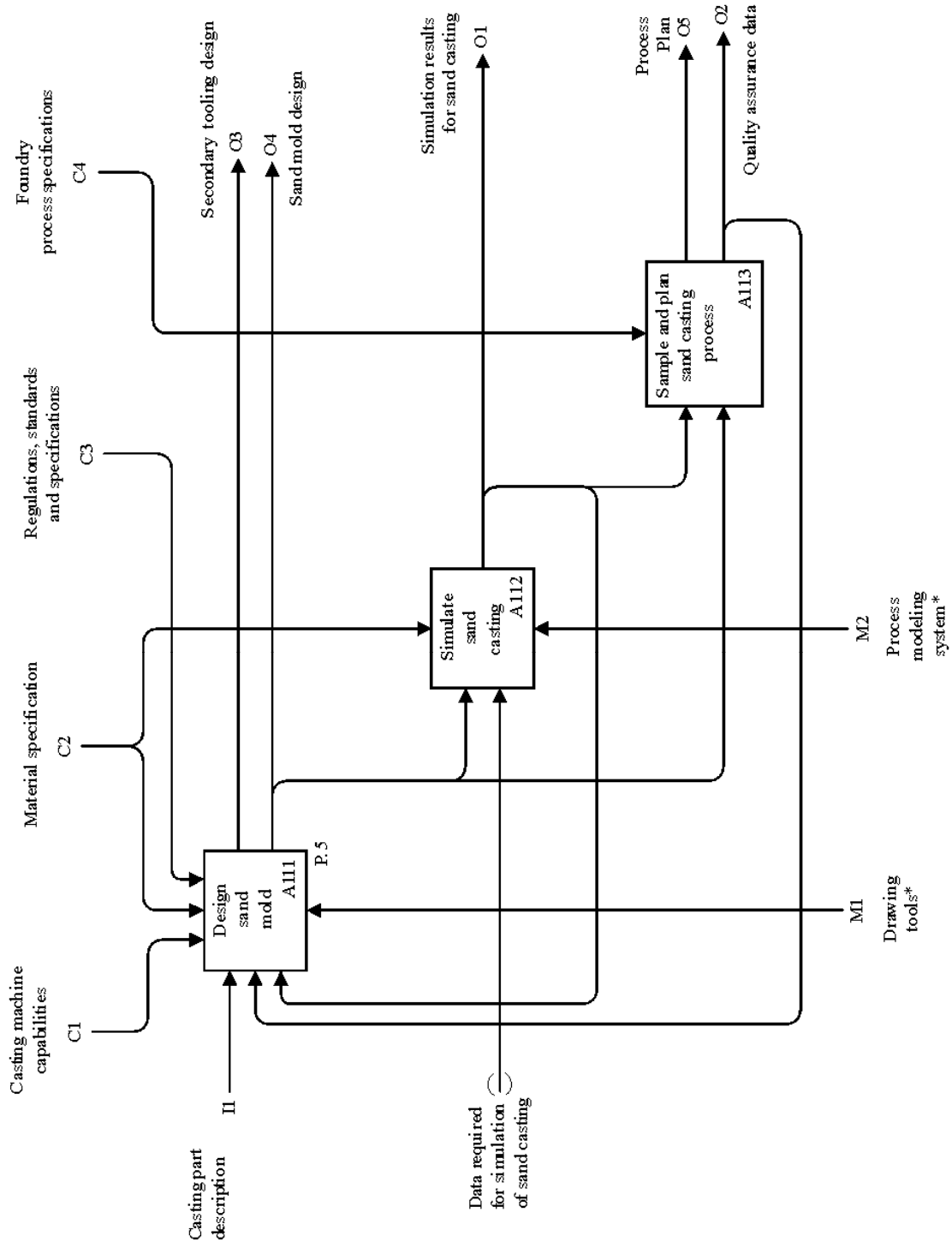


Figure F.5 - A11 Design sand mold and plan process

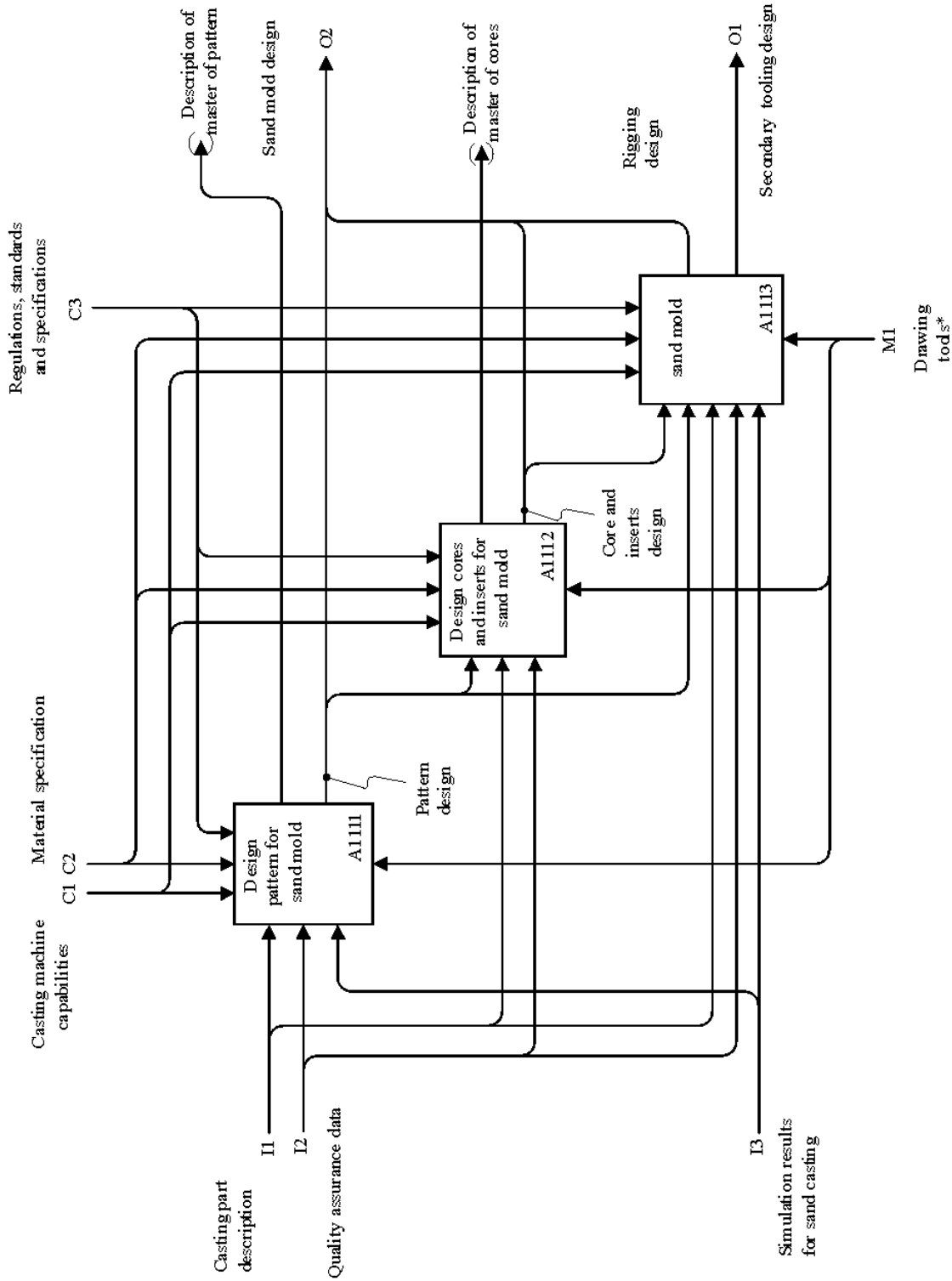


Figure F.6 - A111 Design sand mold

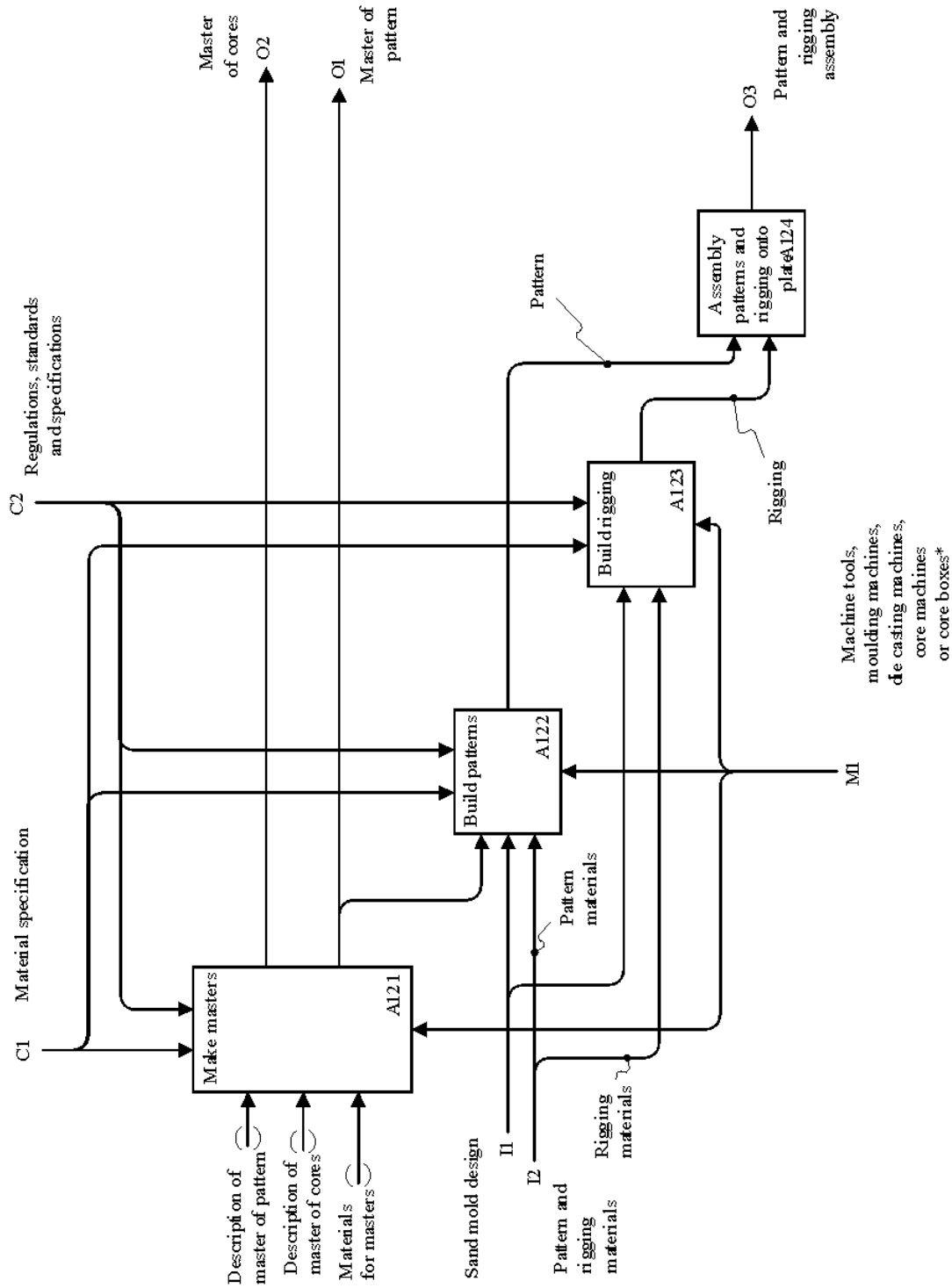


Figure F.7 - A12 Build sand mold



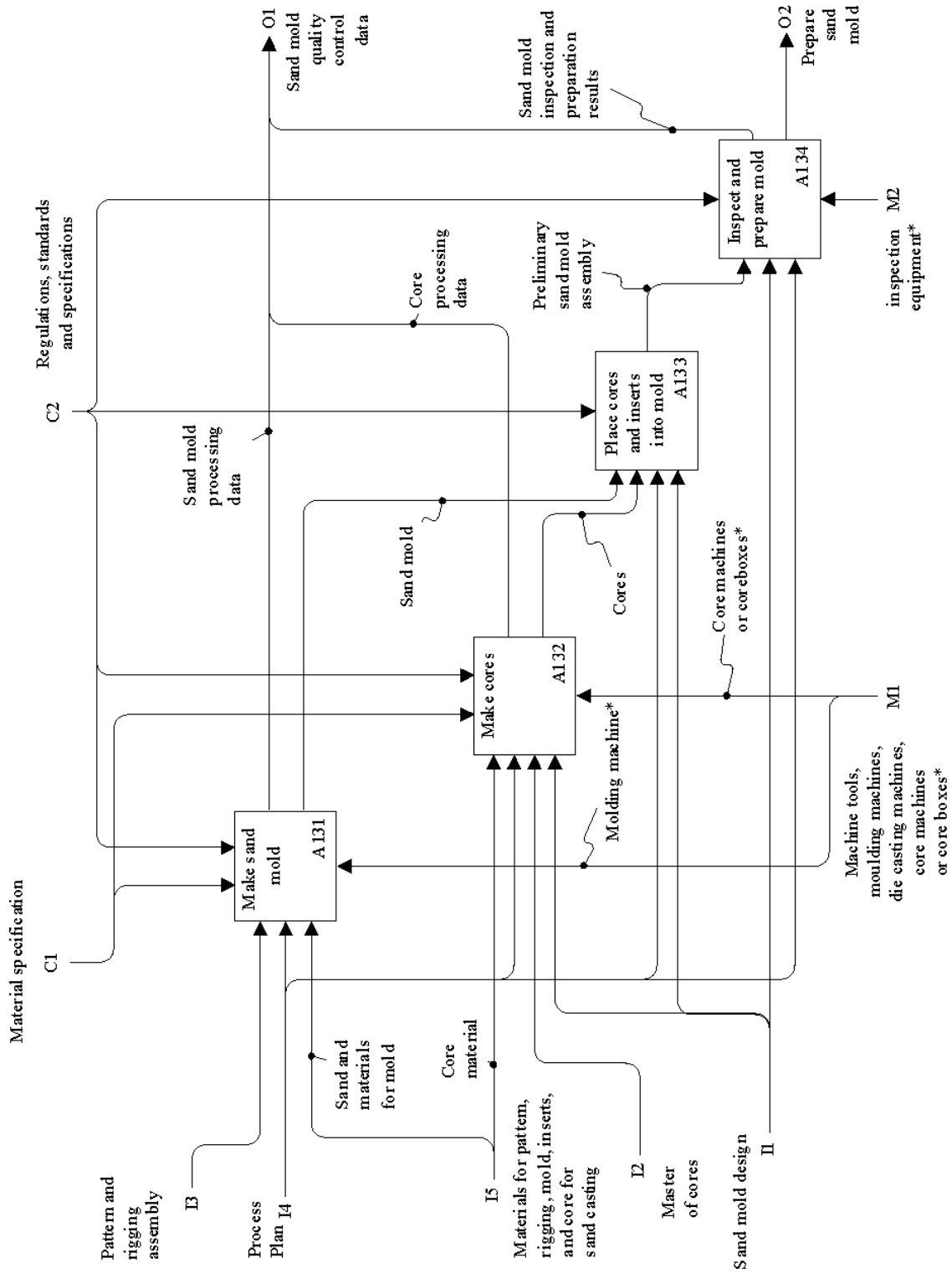


Figure F.8 - A13 Build pattern and rigging assembly

1

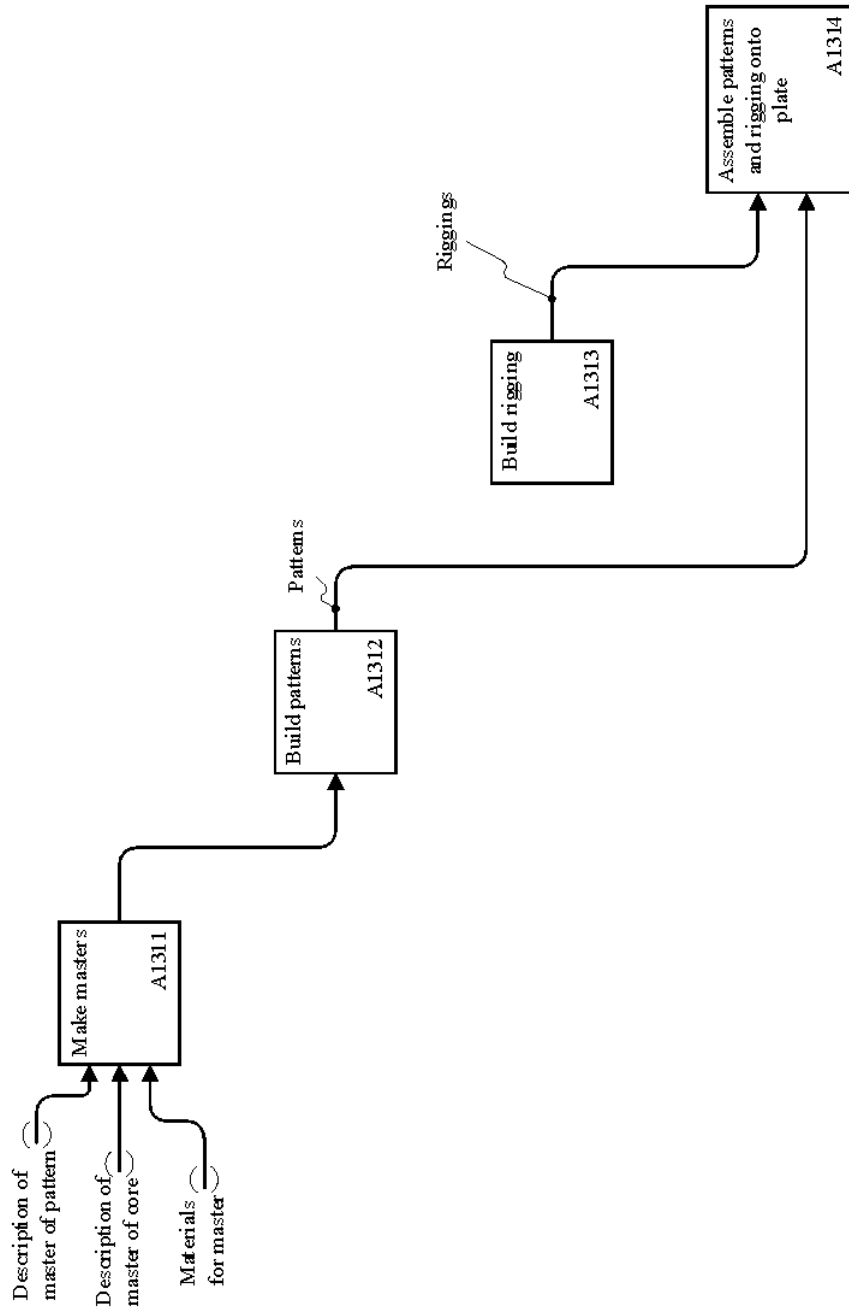


Figure F.9 - A131 Make sand mold

Figure F.10 — A15 Pour cool and shakeout

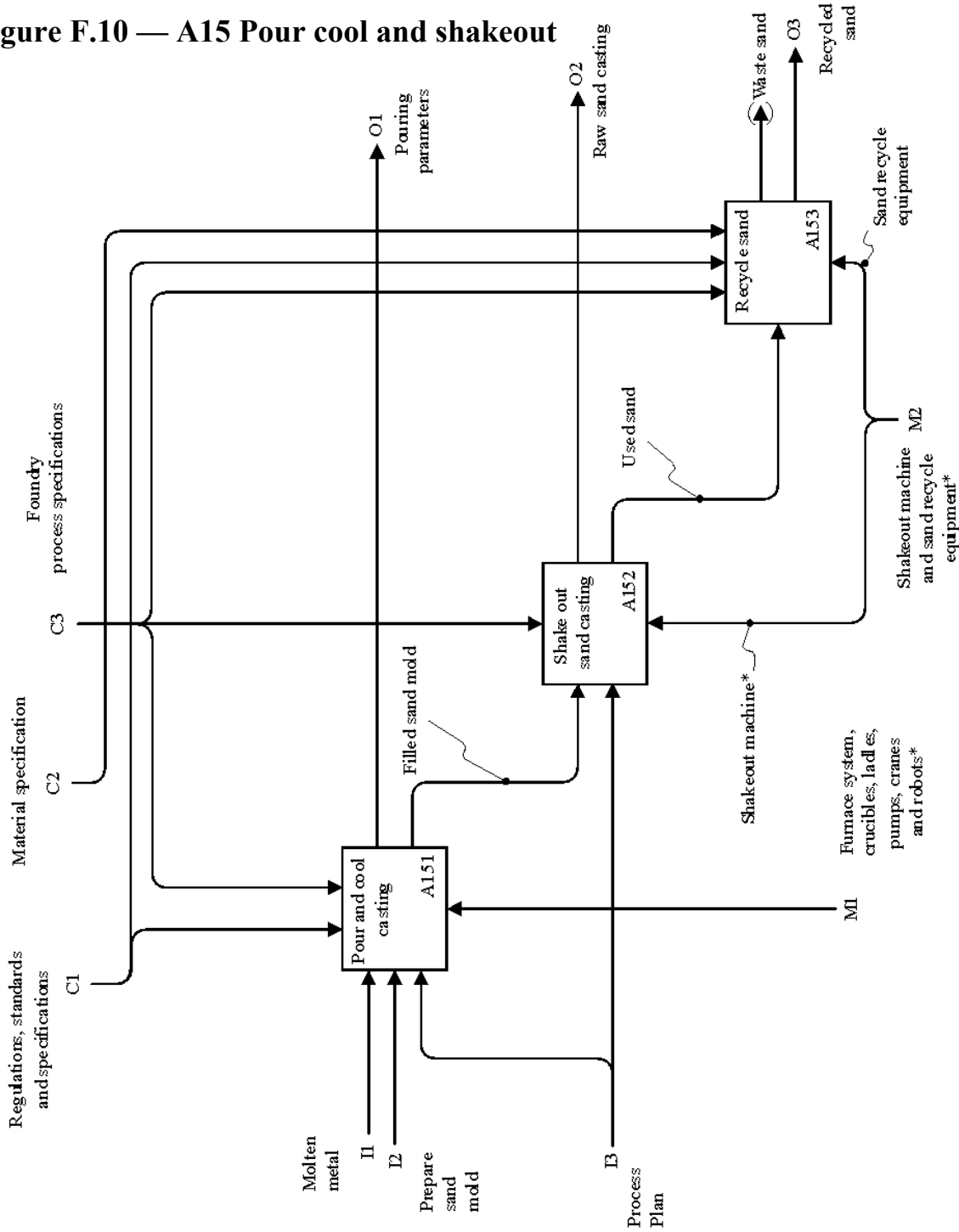


Figure F.10 - A15 Pour, cool and Shakeout

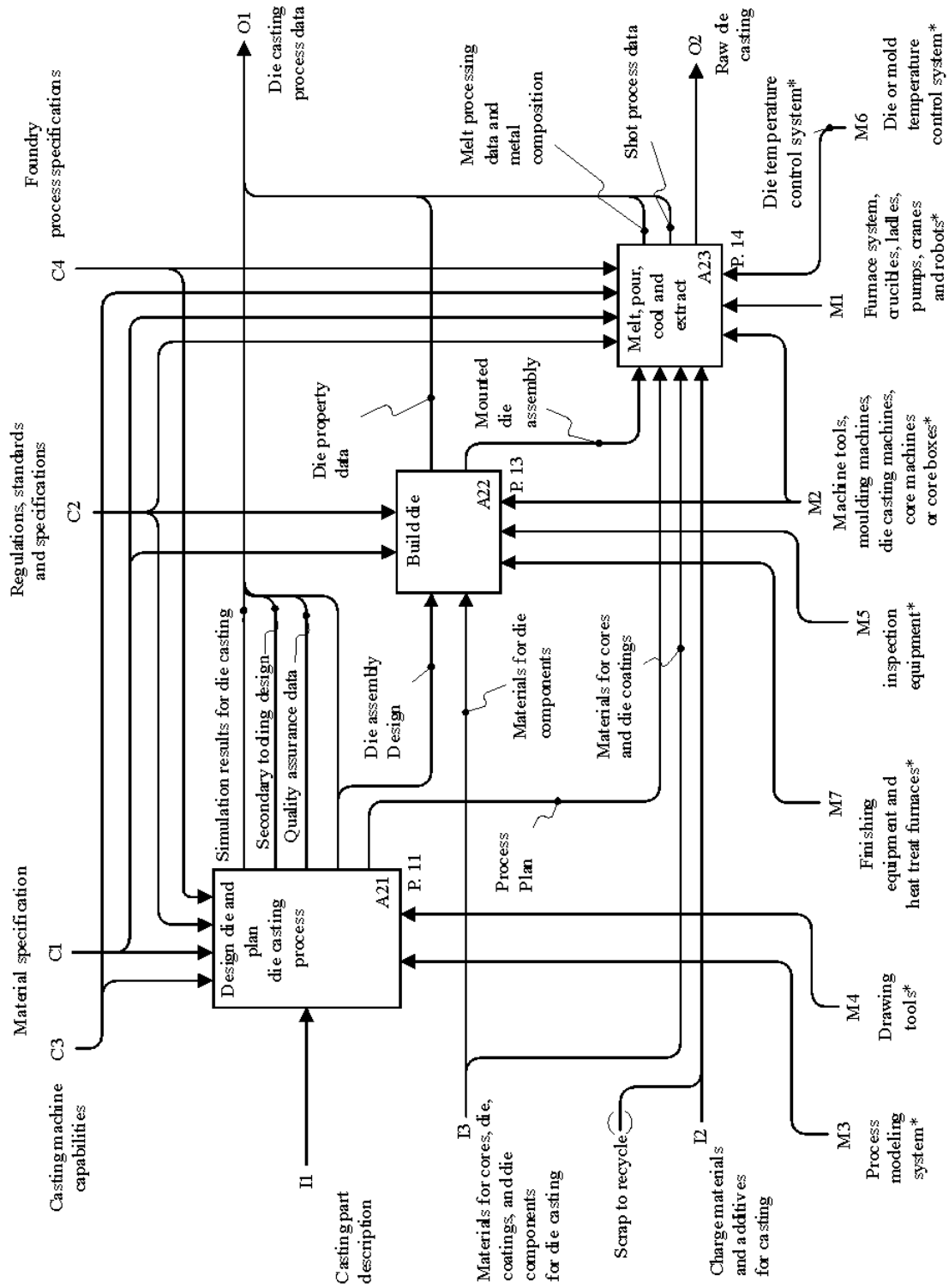


Figure F.11 - A2 Die cast a part

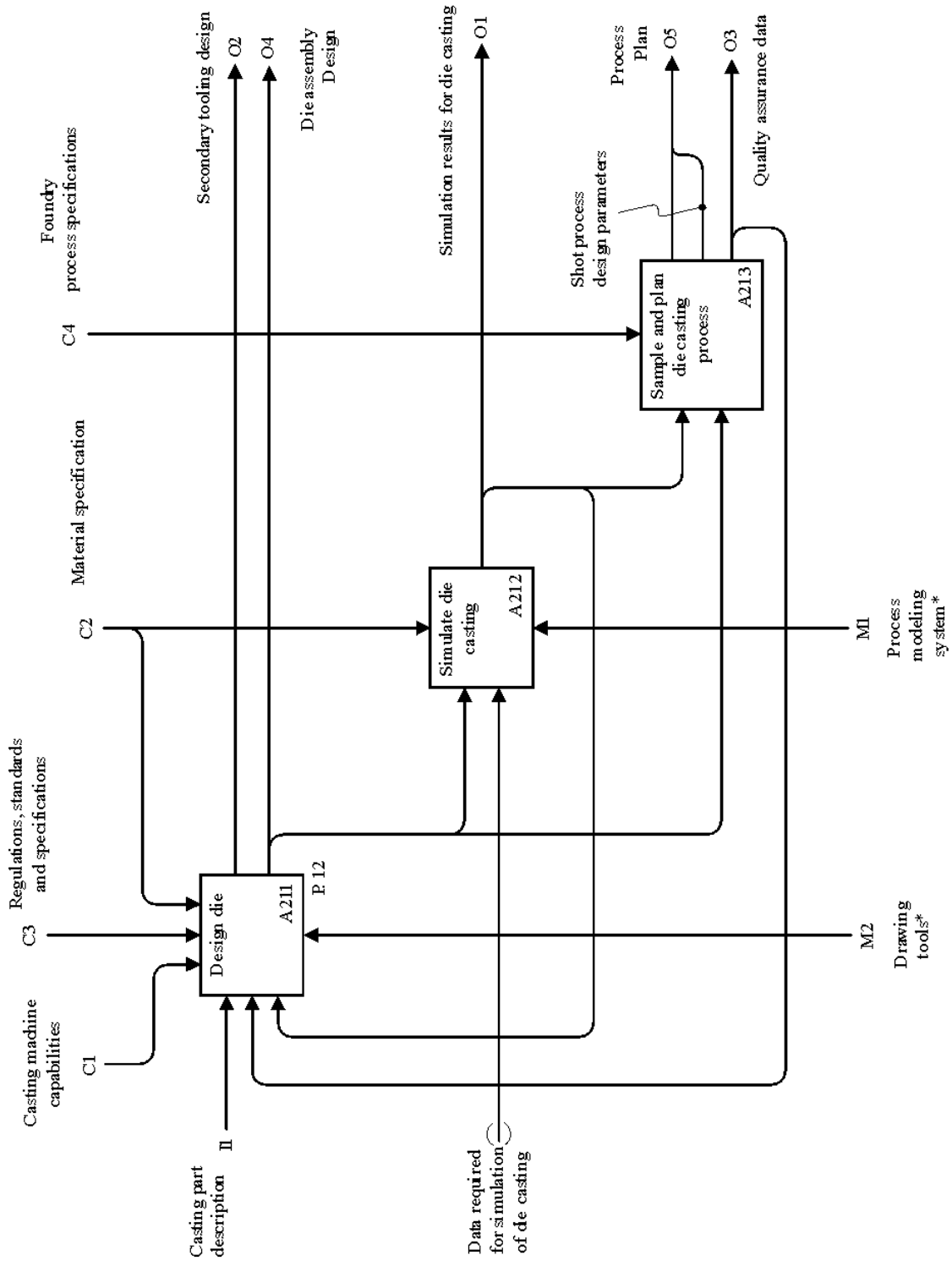


Figure F.12 - A12 Design die and plan die casting process

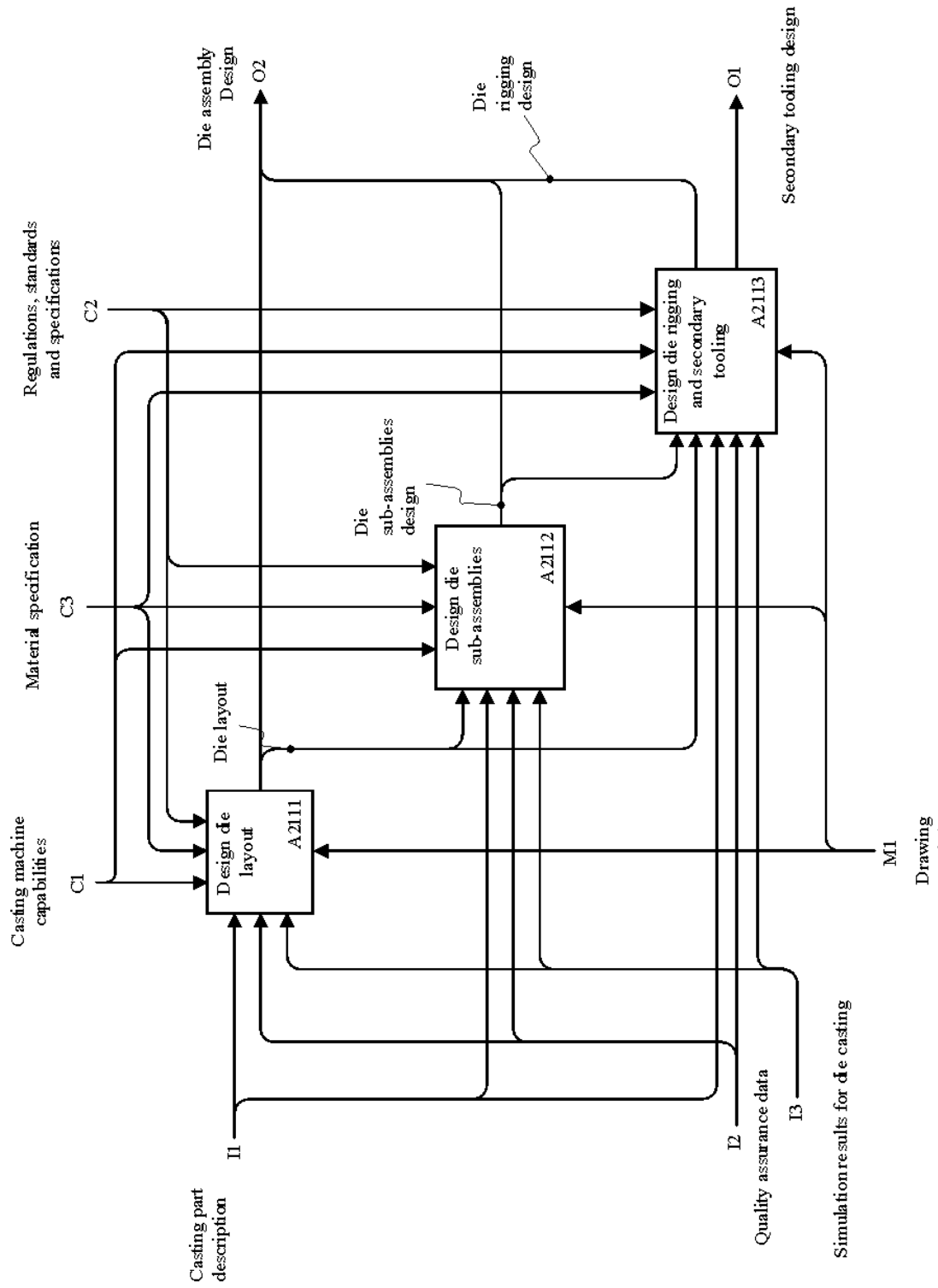


Figure F.13 - A2111 Design die

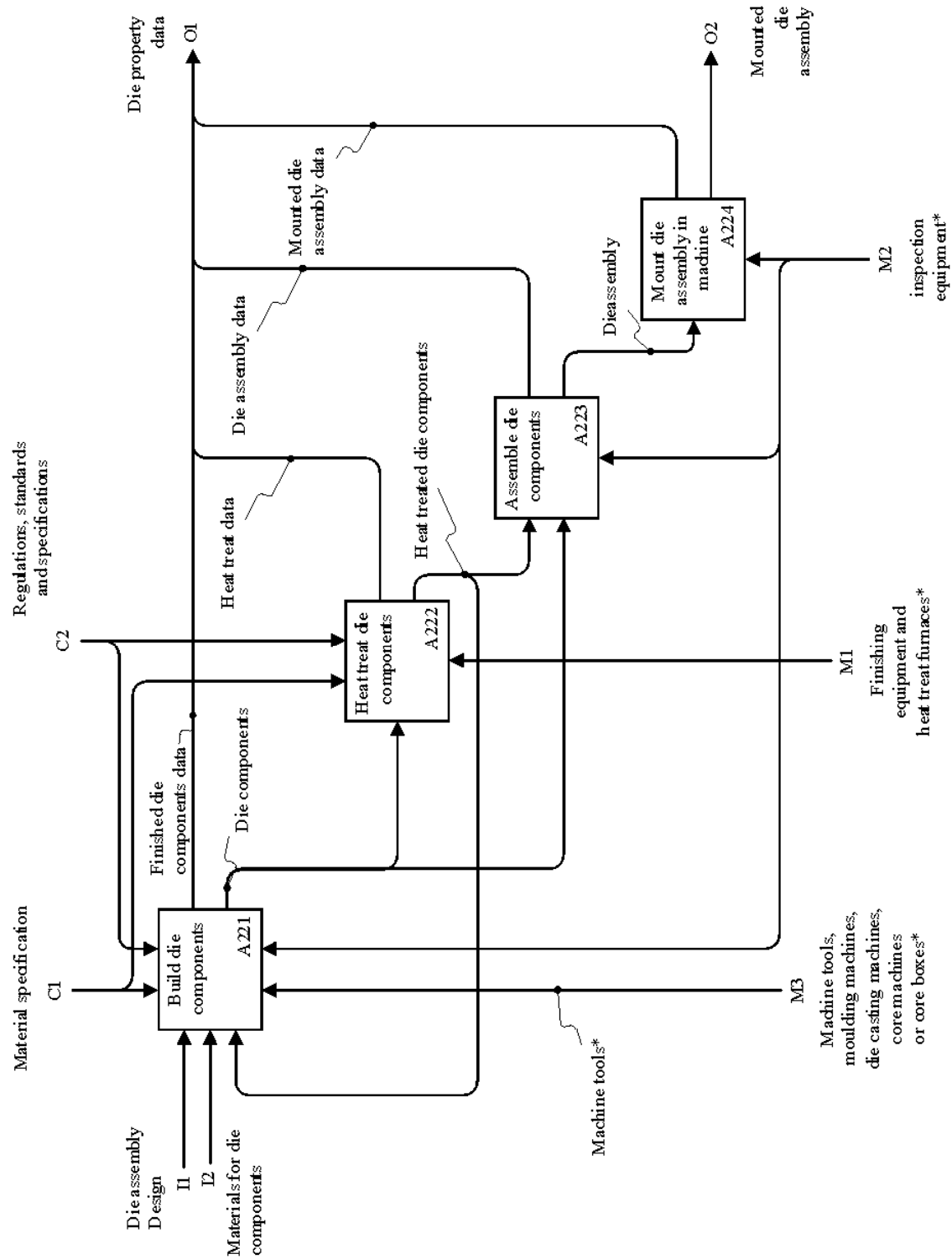


Figure F.14 - A22 Build die

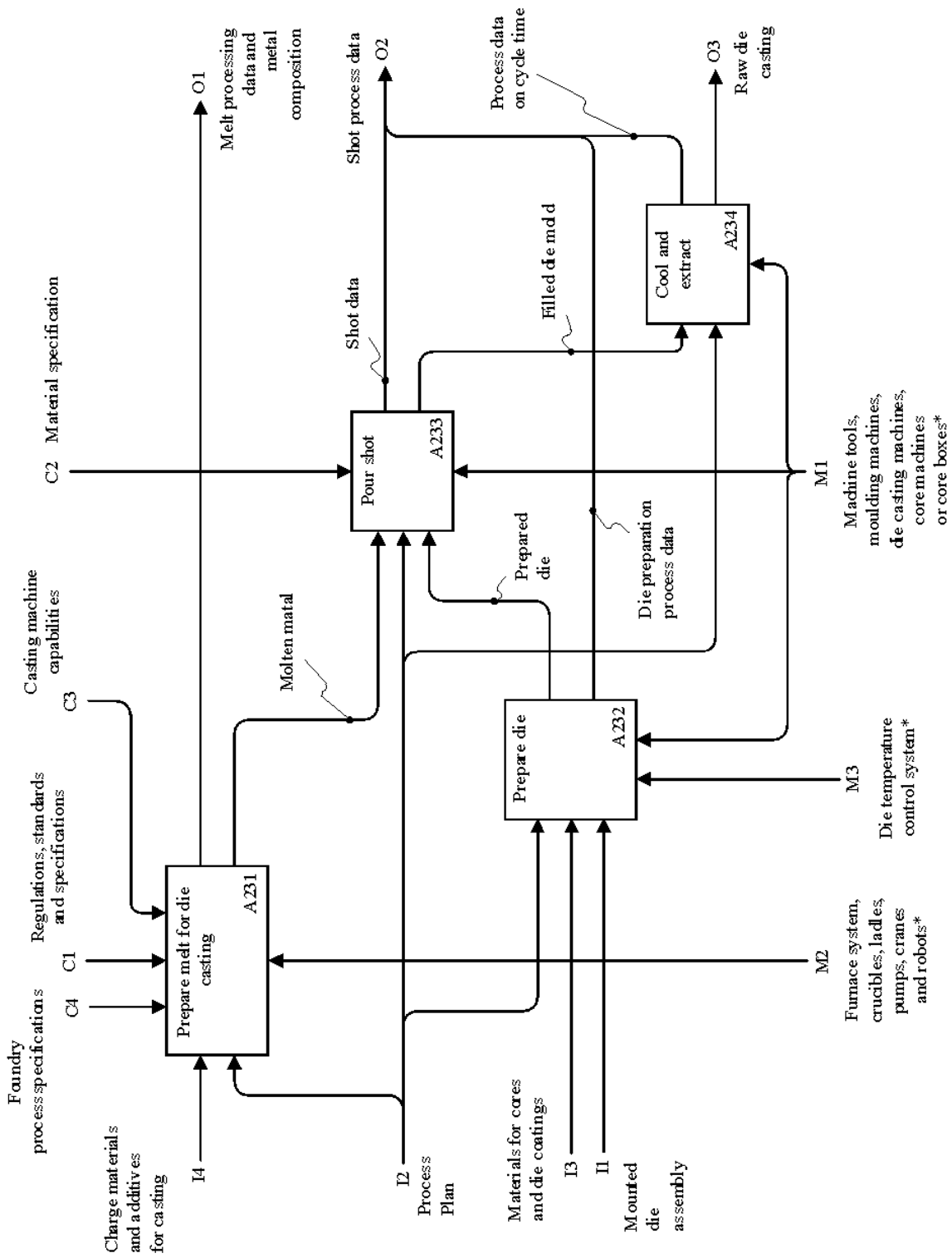


Figure F.15 - A23 Melt, pour, cool and extract





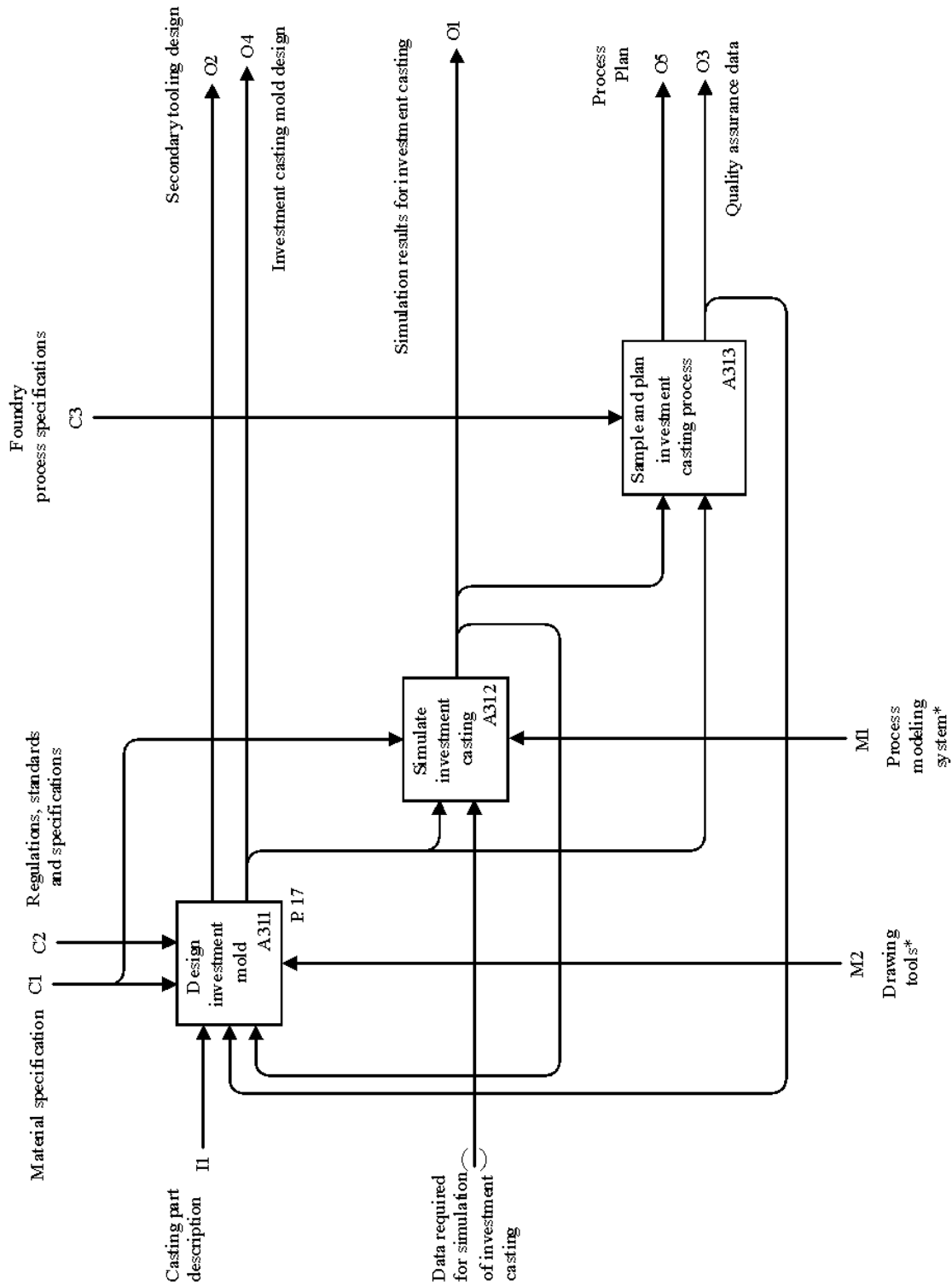


Figure F.17 - A31 Design investment pattern assembly and plan process

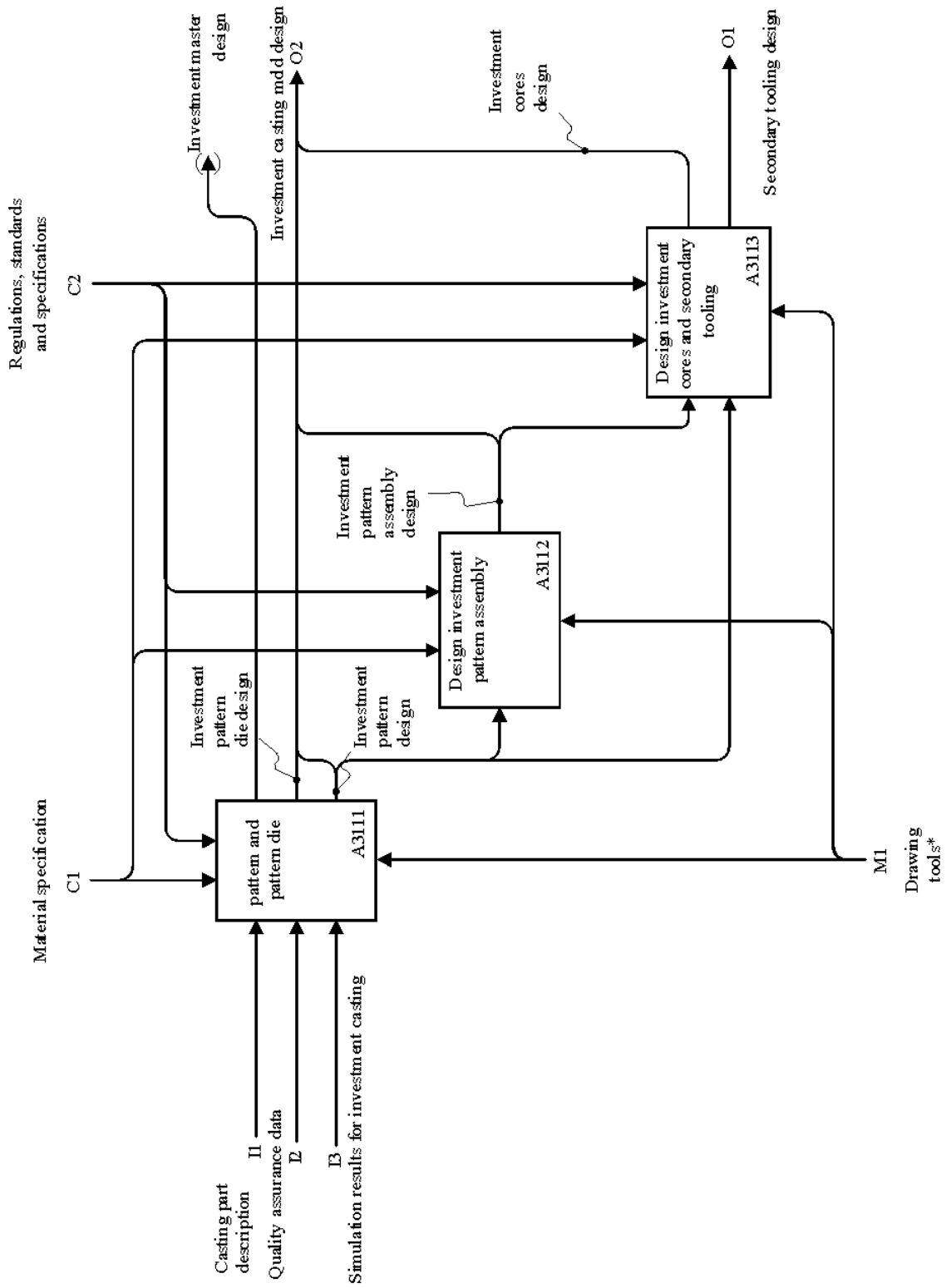


Figure F.18 - A311 Design investment mold

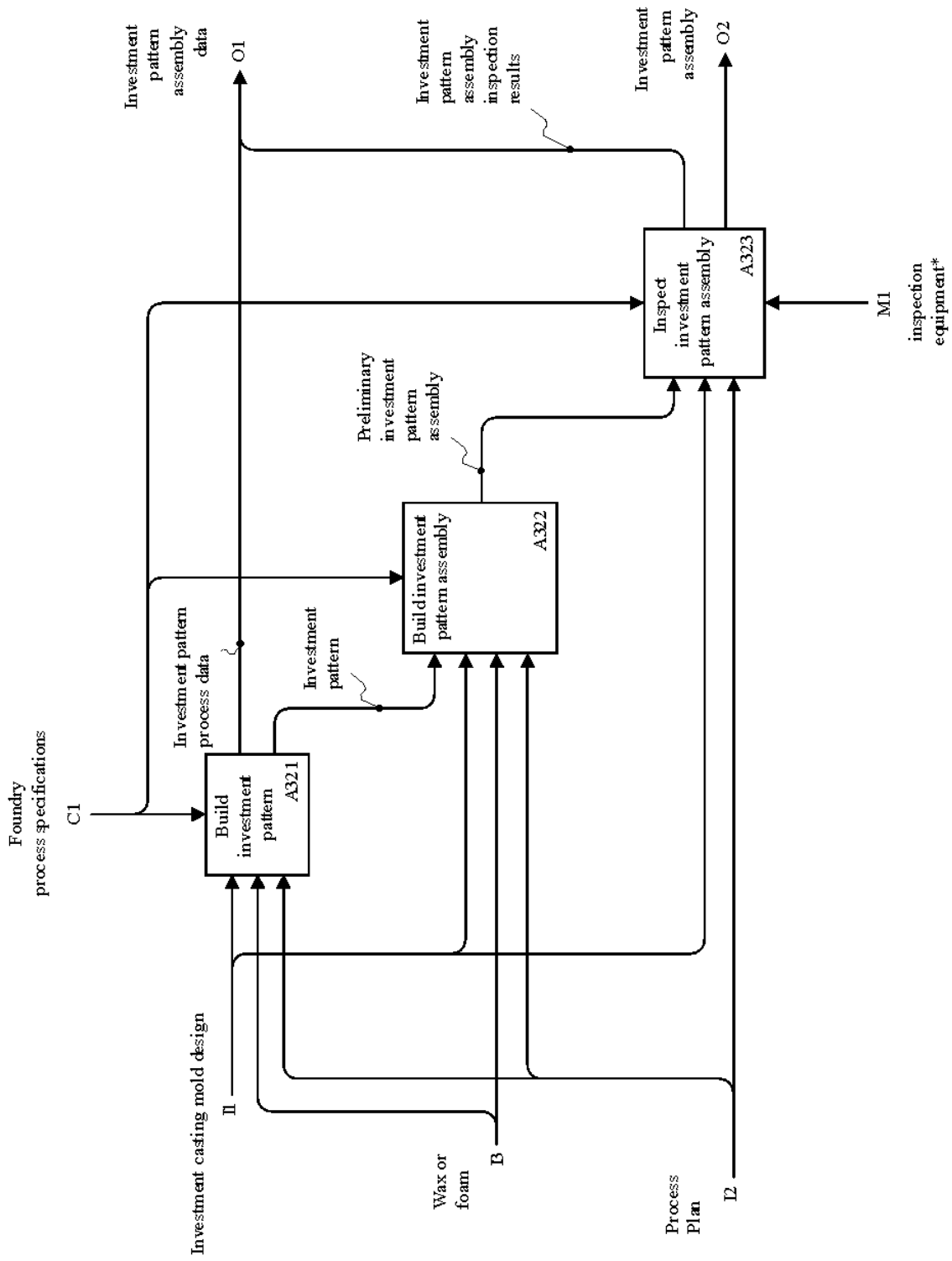


Figure F.19 - A32 Build wax or foam pattern

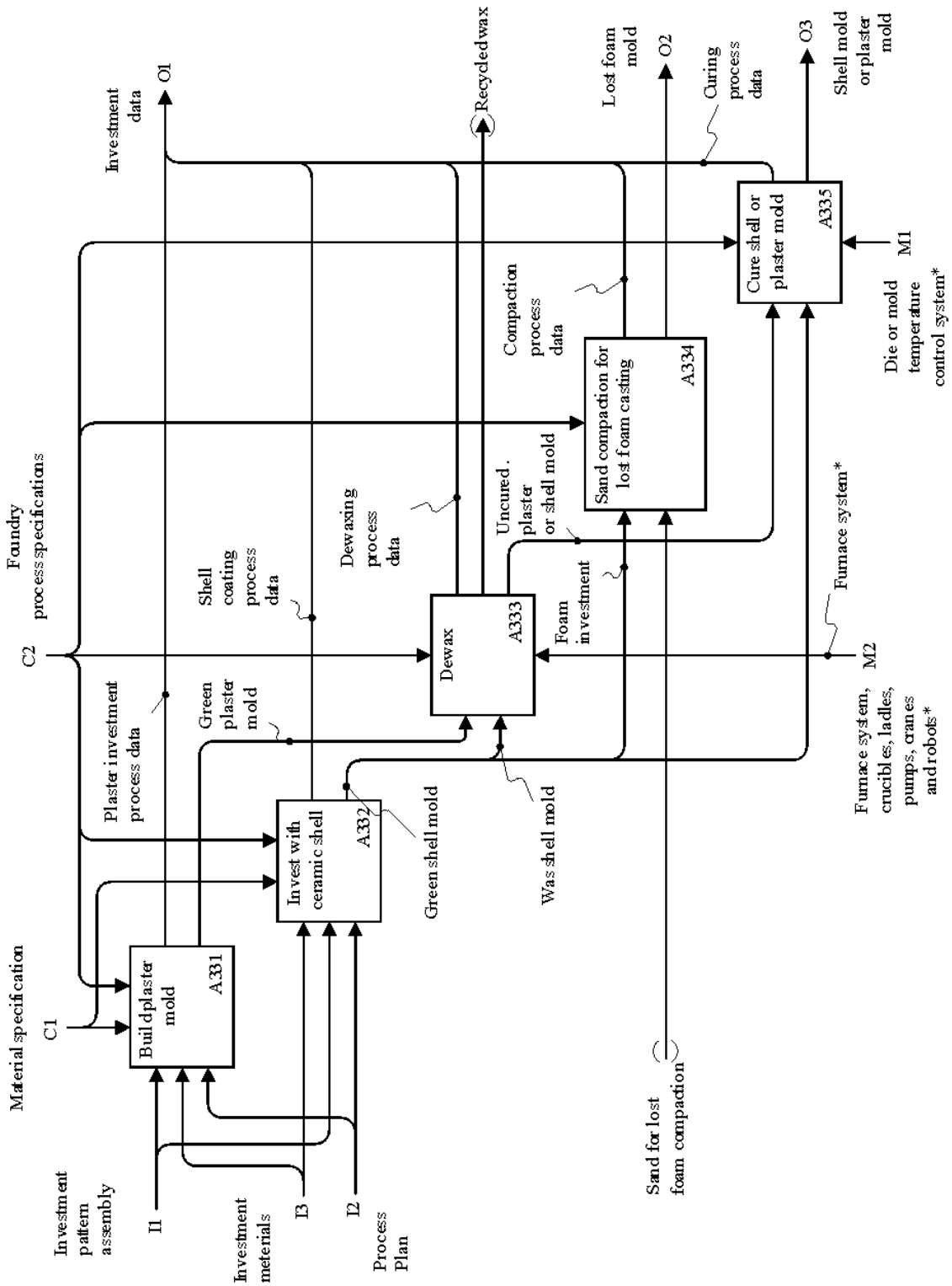


Figure F.20 - A33 Invest pattern

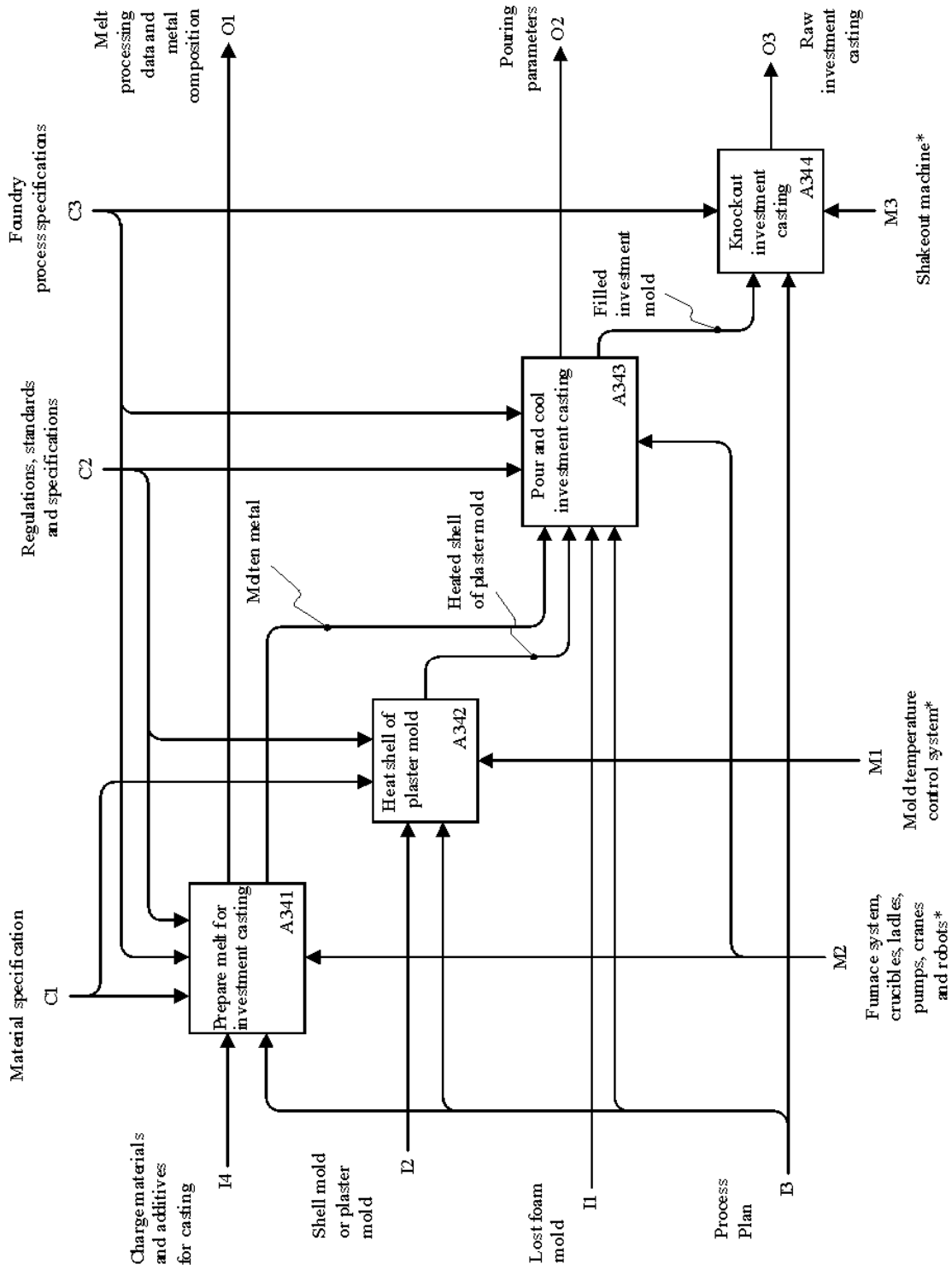


Figure F.21 - A34 Melt, pour, cool and Knockout

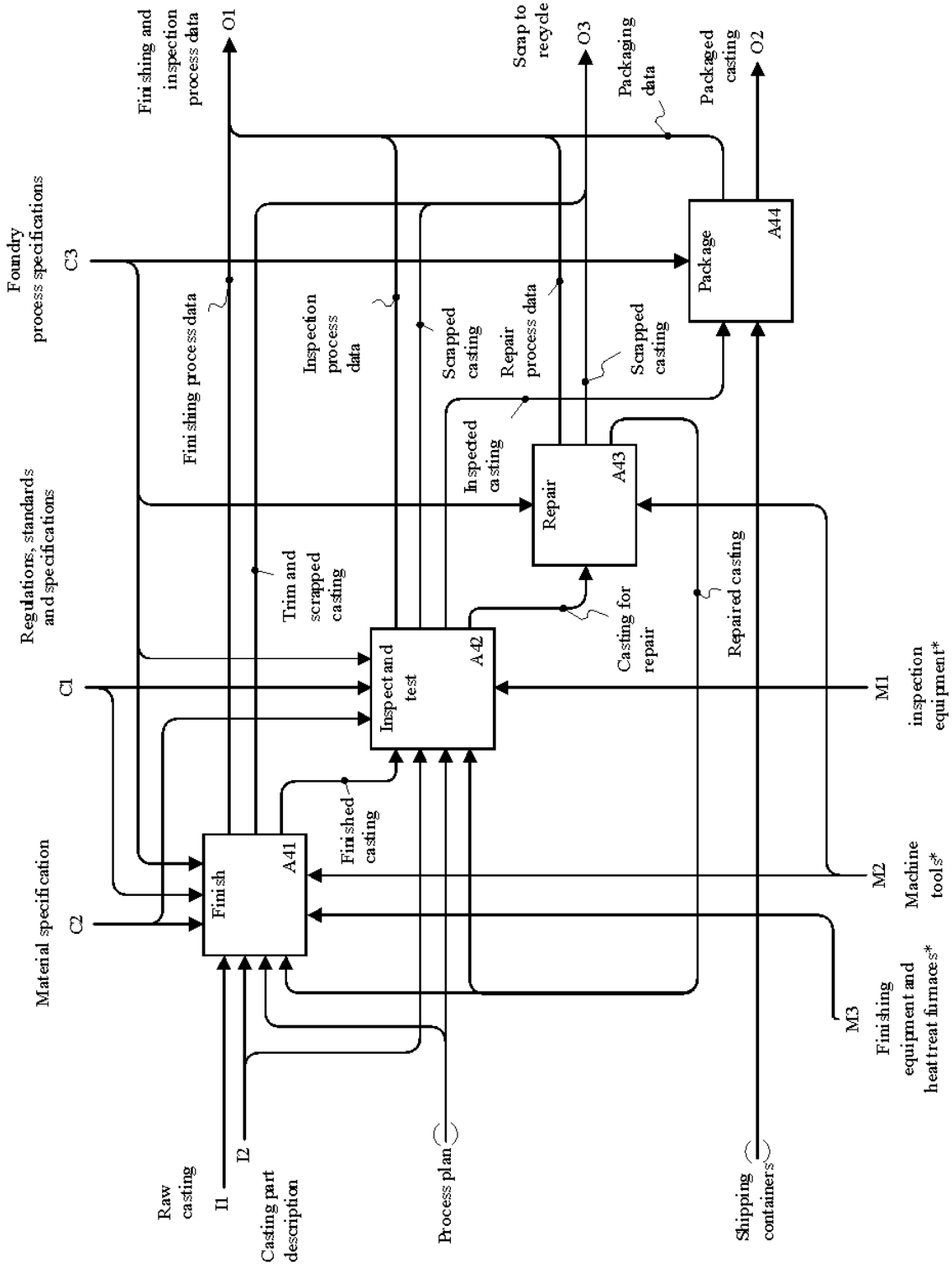


Figure F.22 - A4 Finish and inspect

**Annex G**  
**(informative)**

**Application reference model**

This annex provides the application reference model for this part of ISO 10303 and is given in figures G.1 through G.32. The application reference model is a graphical representation of the structure and constraints of the application objects specified in Clause 4. The graphical form of the application reference model is presented in the EXPRESS-G. The application reference model is independent from any implementation method. EXPRESS-G is defined in Annex D of ISO 10303-11:2004.



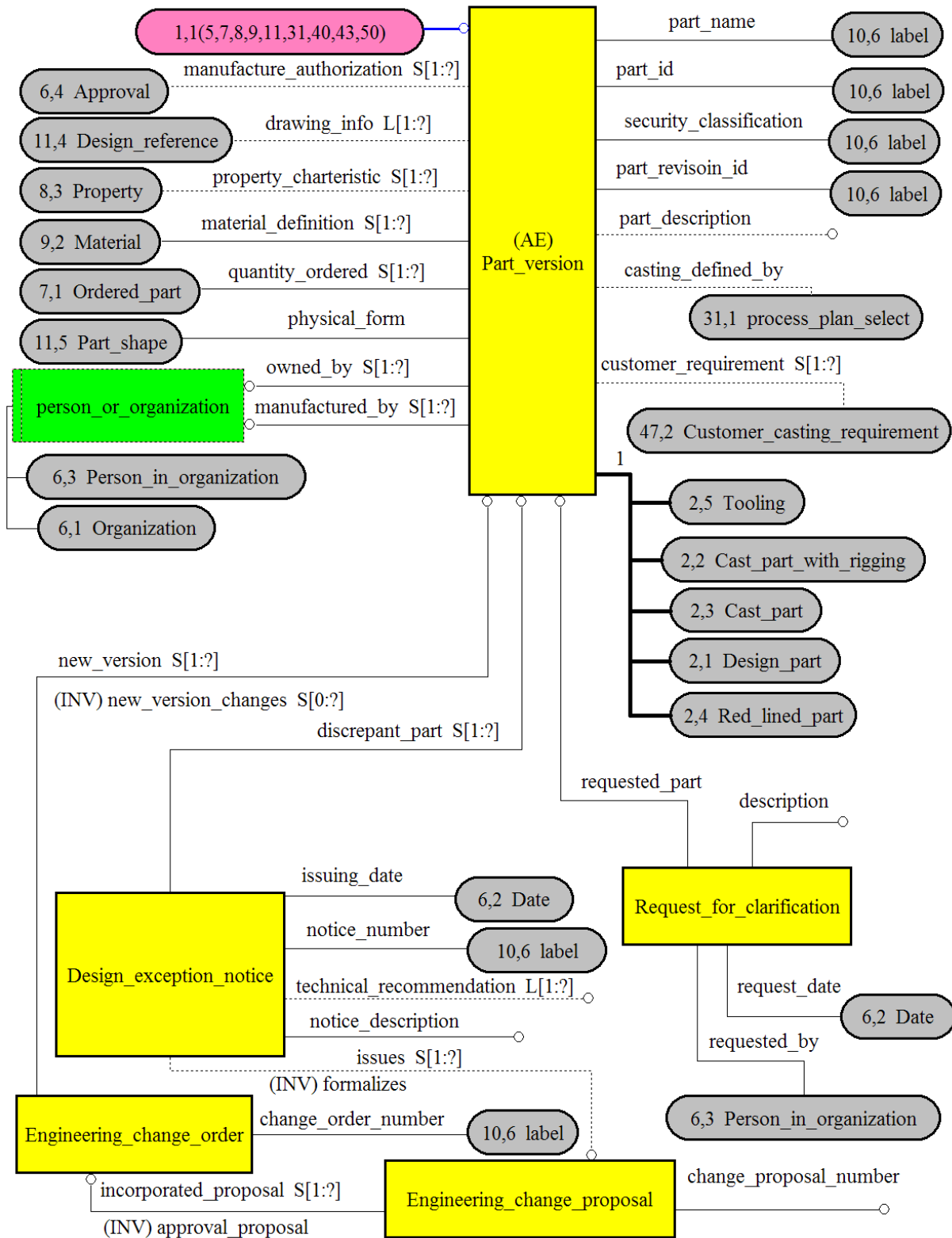


Figure G.1 — ARM diagram (1 of 53)

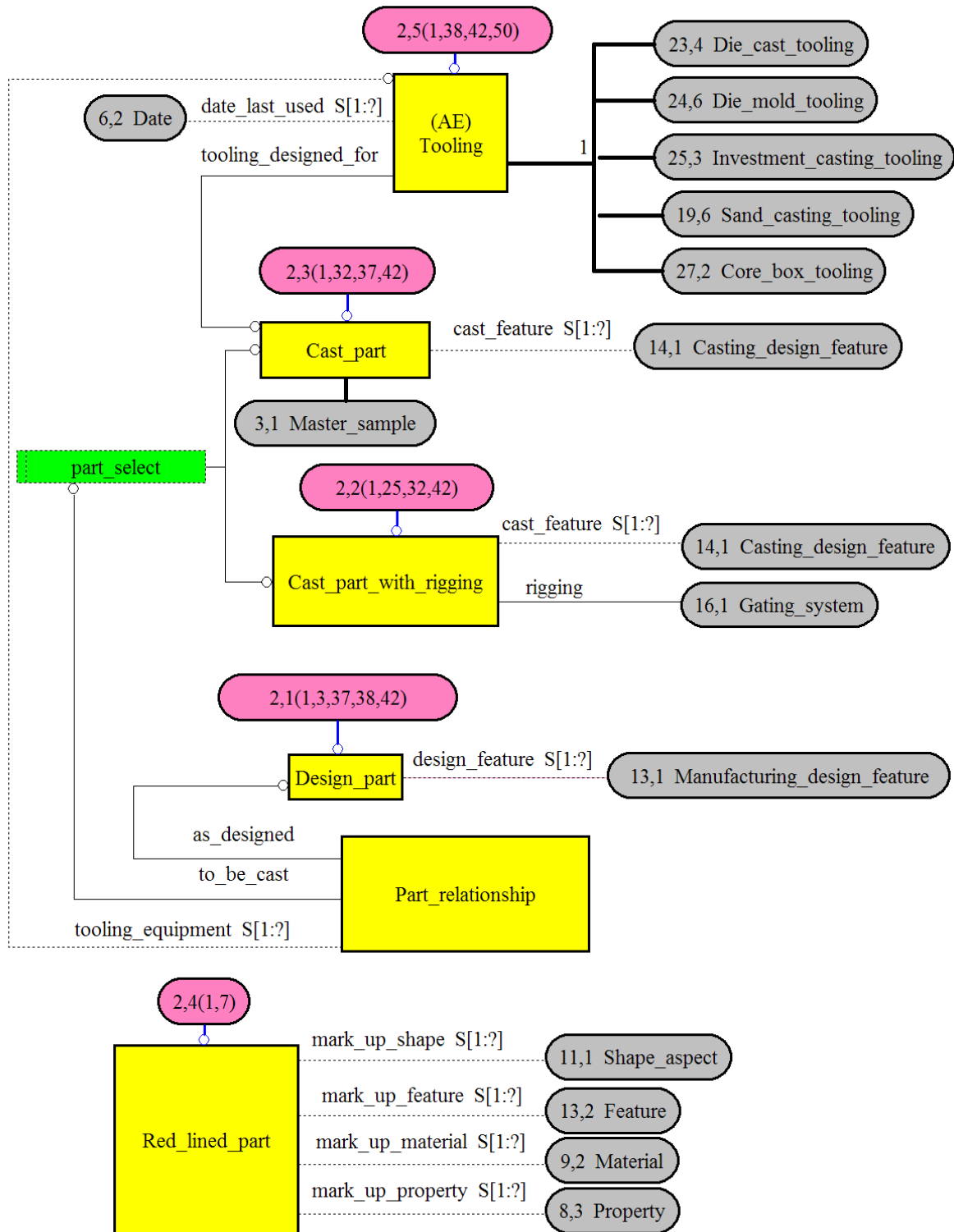
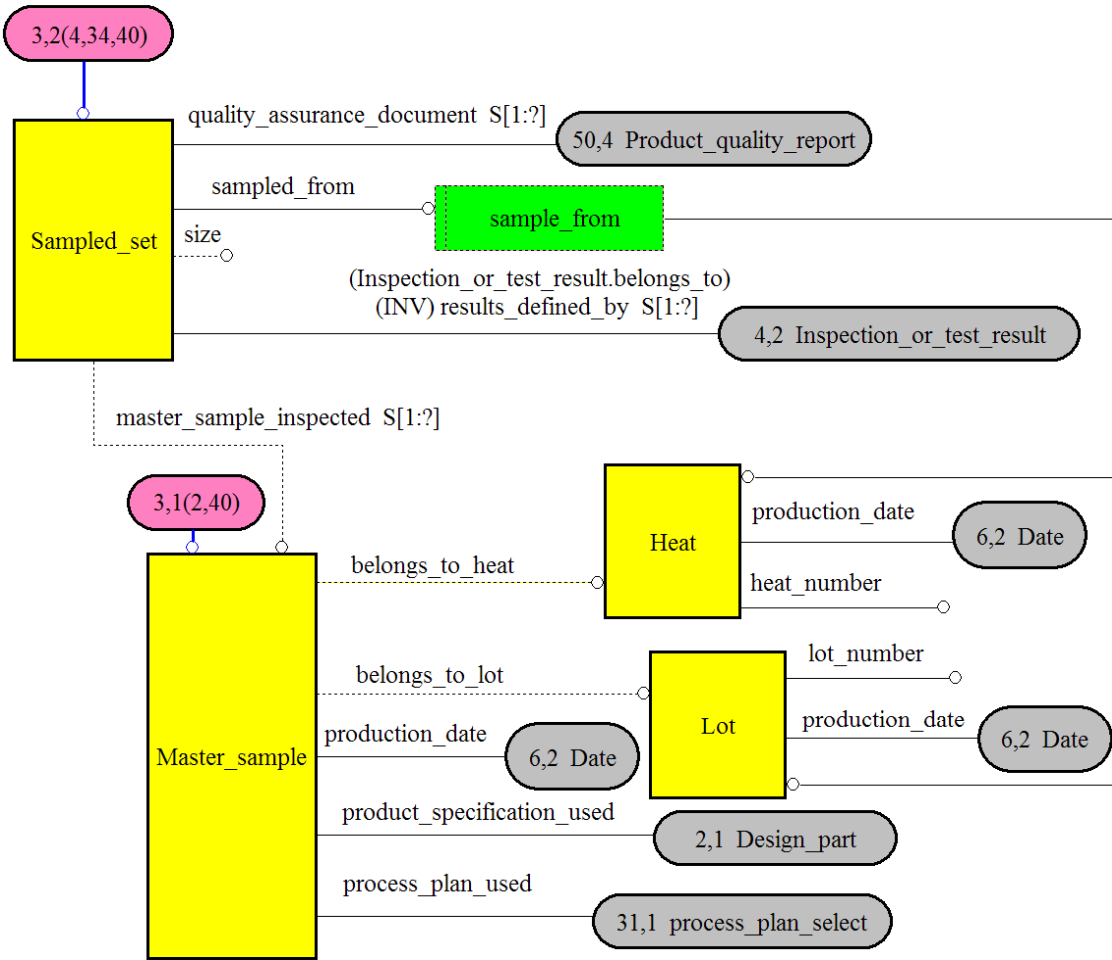


Figure G.2 — ARM diagram (2 of 53)



**Figure G.3 — ARM diagram (3 of 53)**

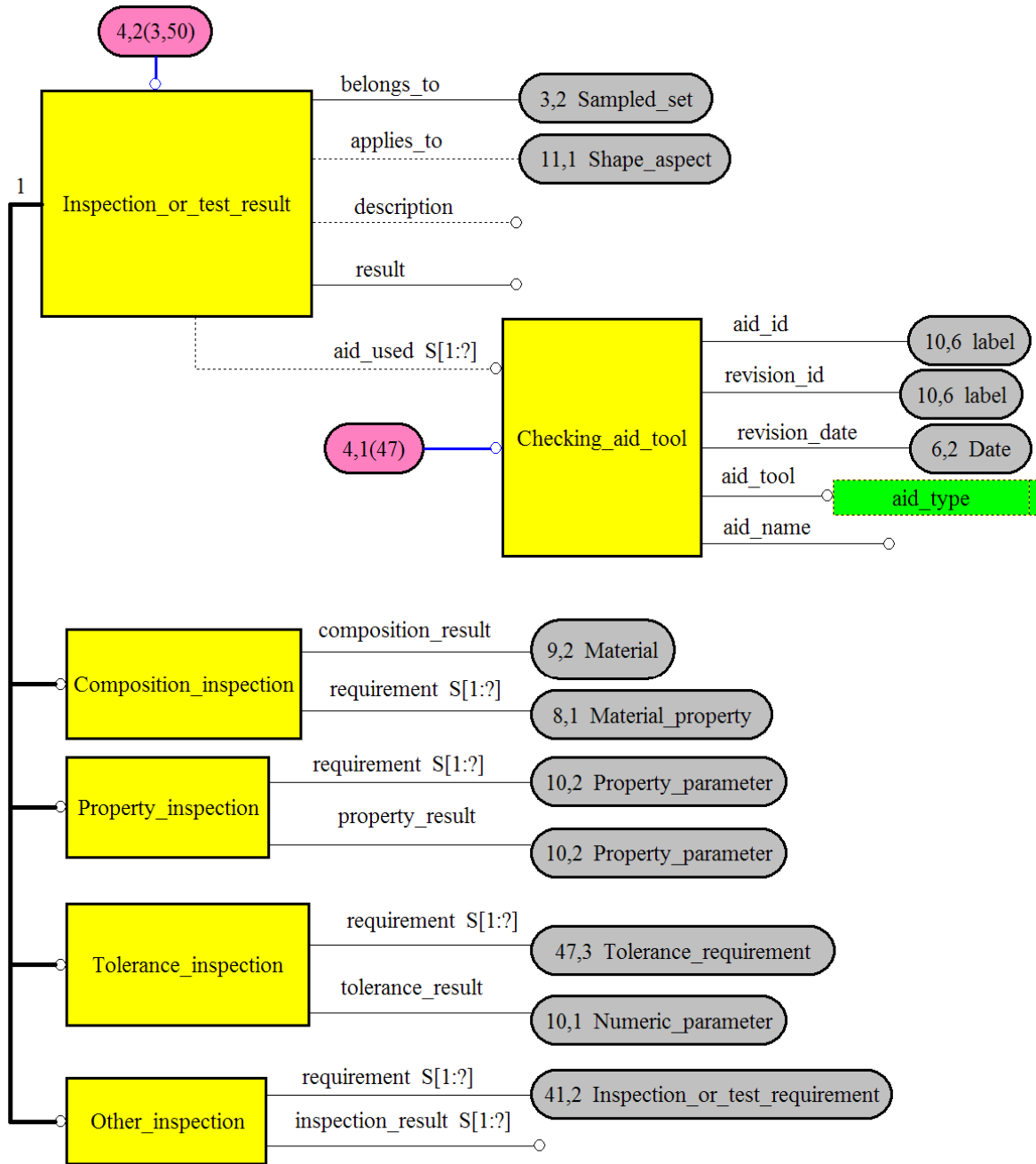


Figure G.4 — ARM diagram (4 of 53)

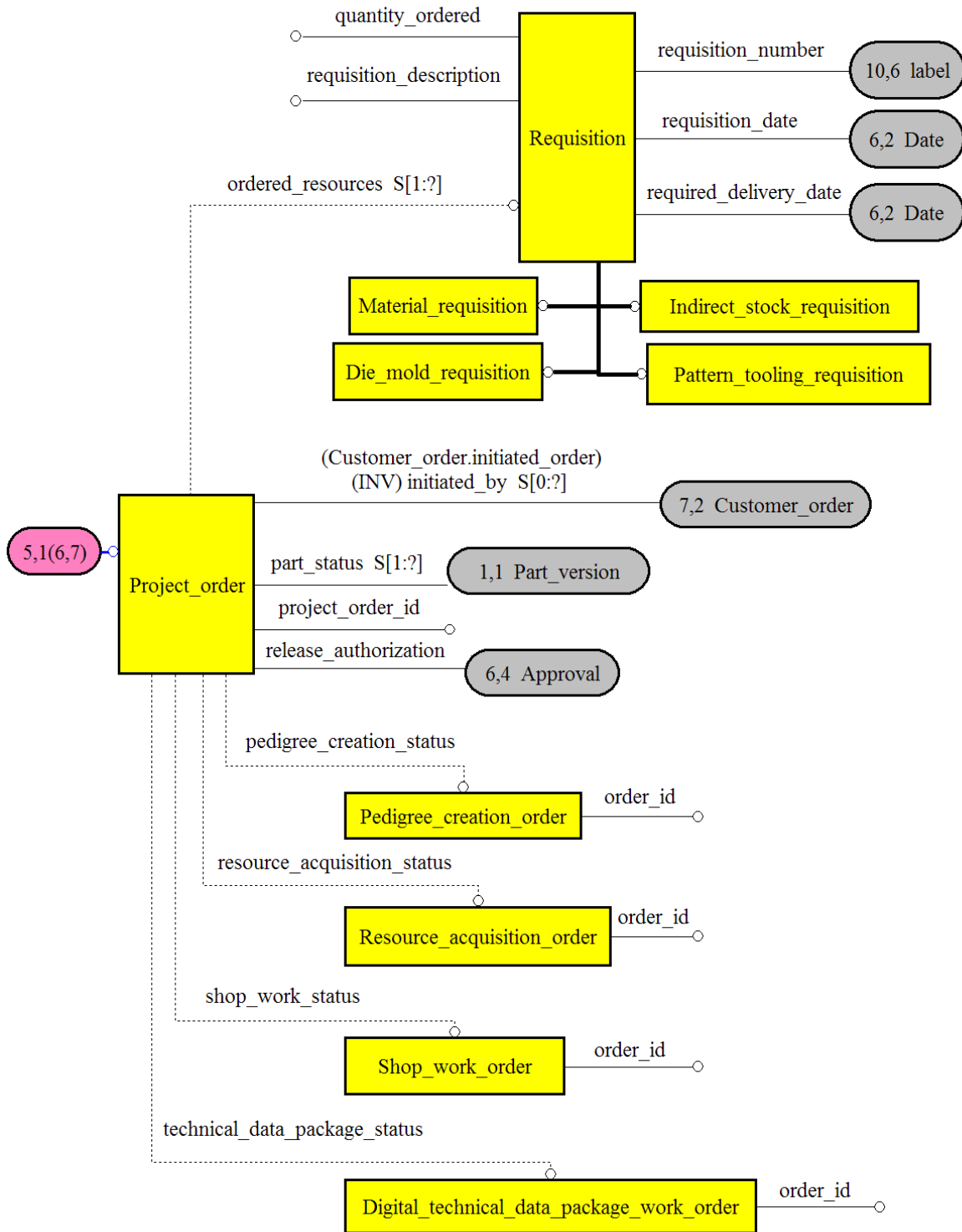


Figure G.5 — ARM diagram (5 of 53)

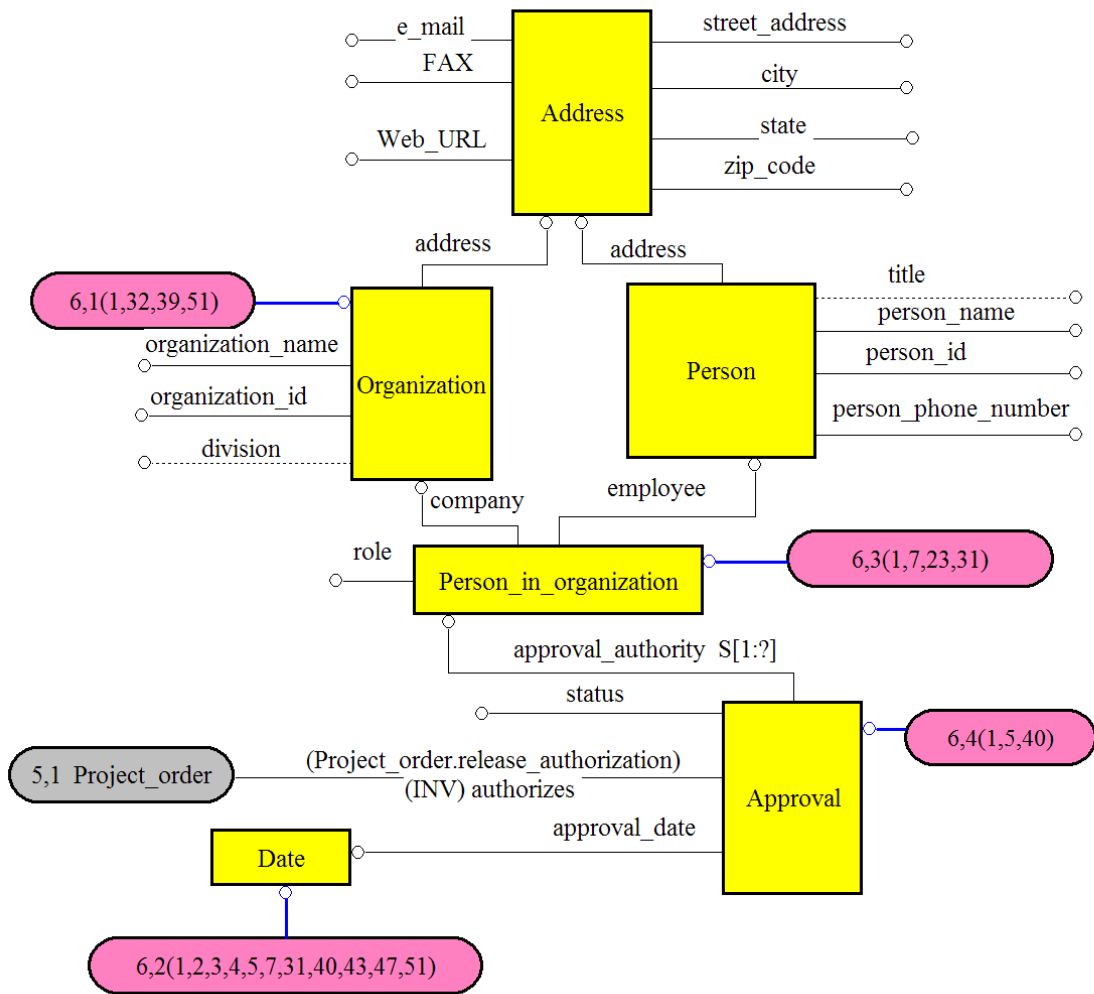


Figure G.6 — ARM diagram (6 of 53)

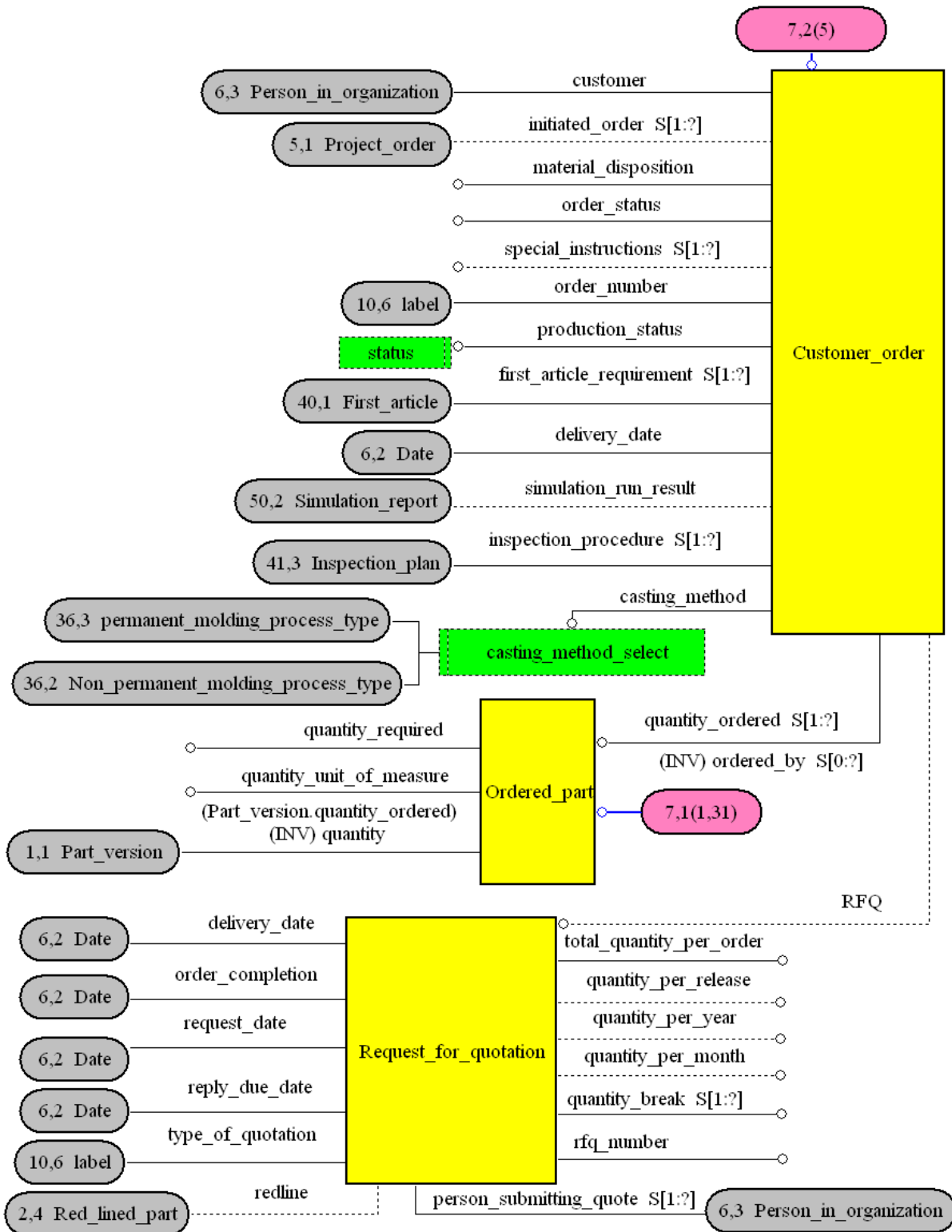


Figure G.7 — ARM diagram (7 of 53)

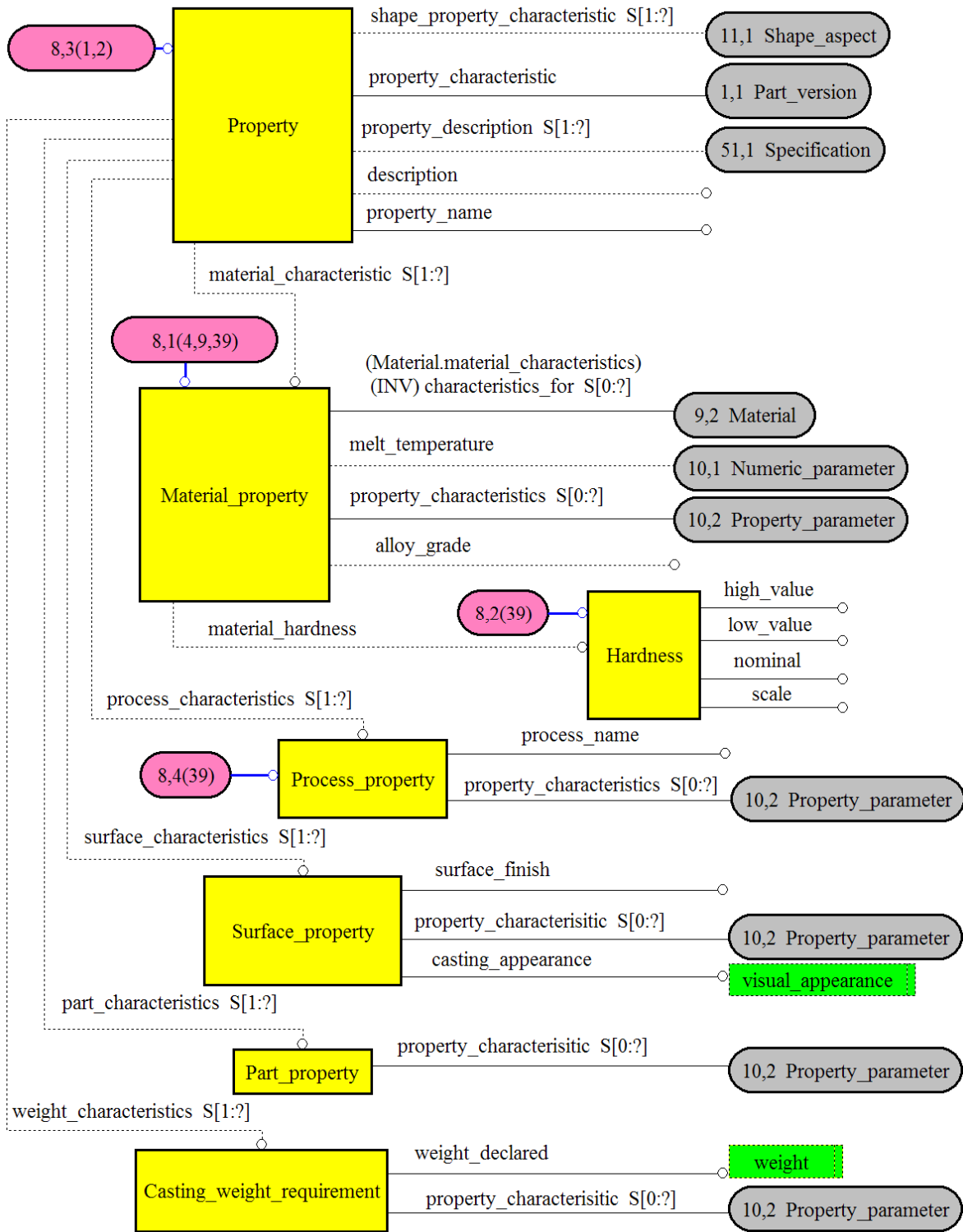


Figure G.8 — ARM diagram (8 of 53)



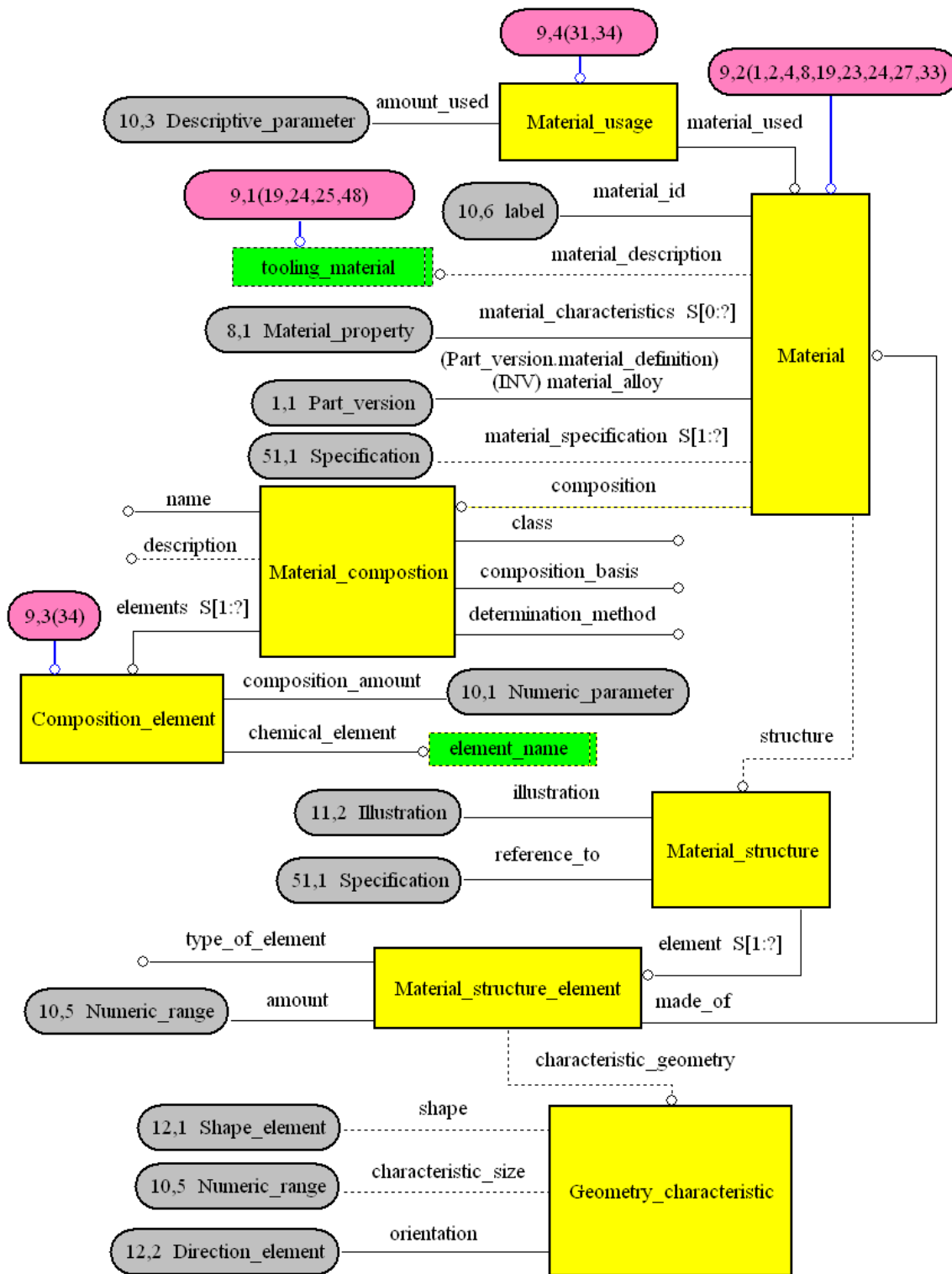


Figure G.9 — ARM diagram (9 of 53)

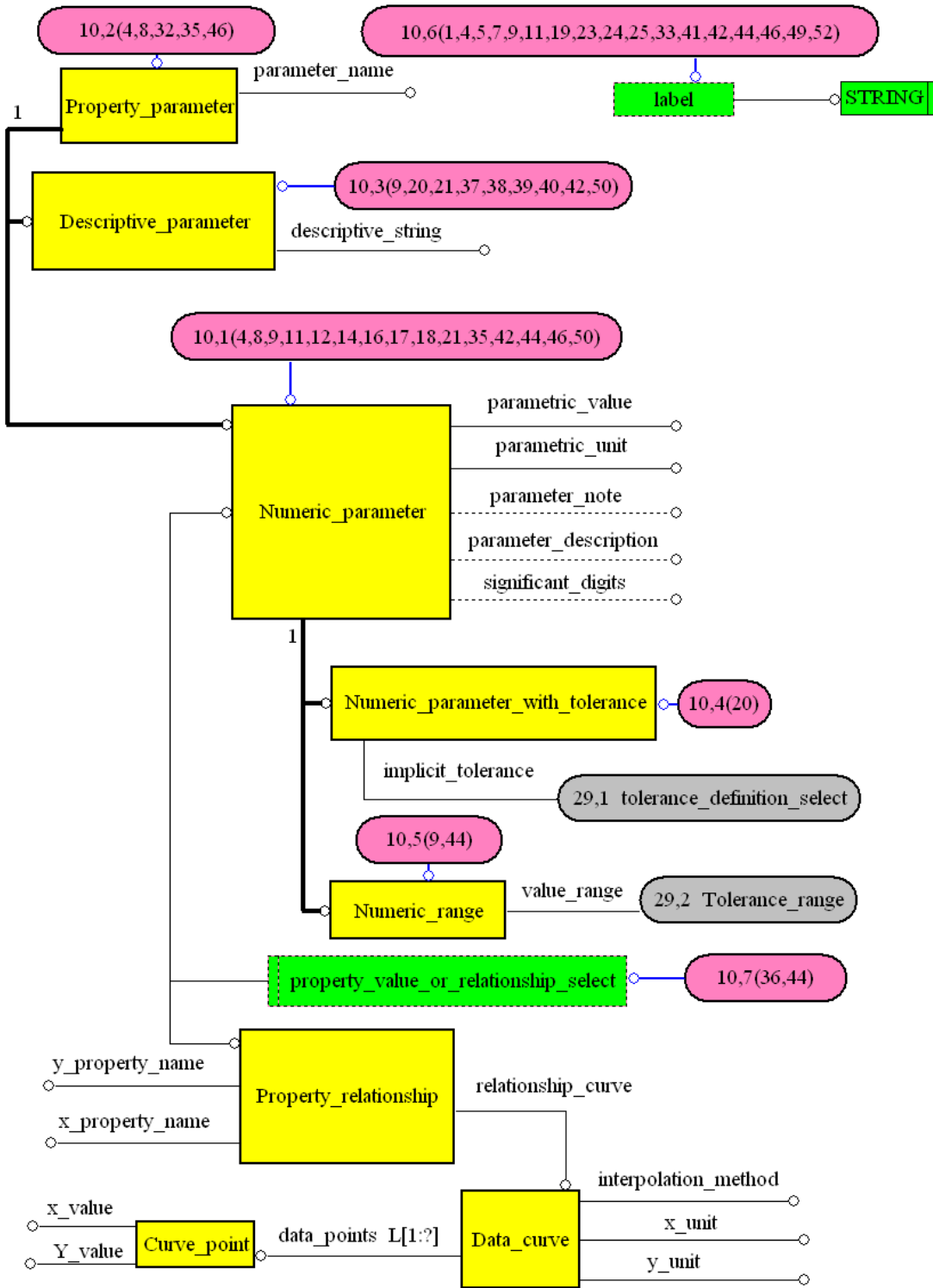


Figure G.10 — ARM diagram (10 of 53)

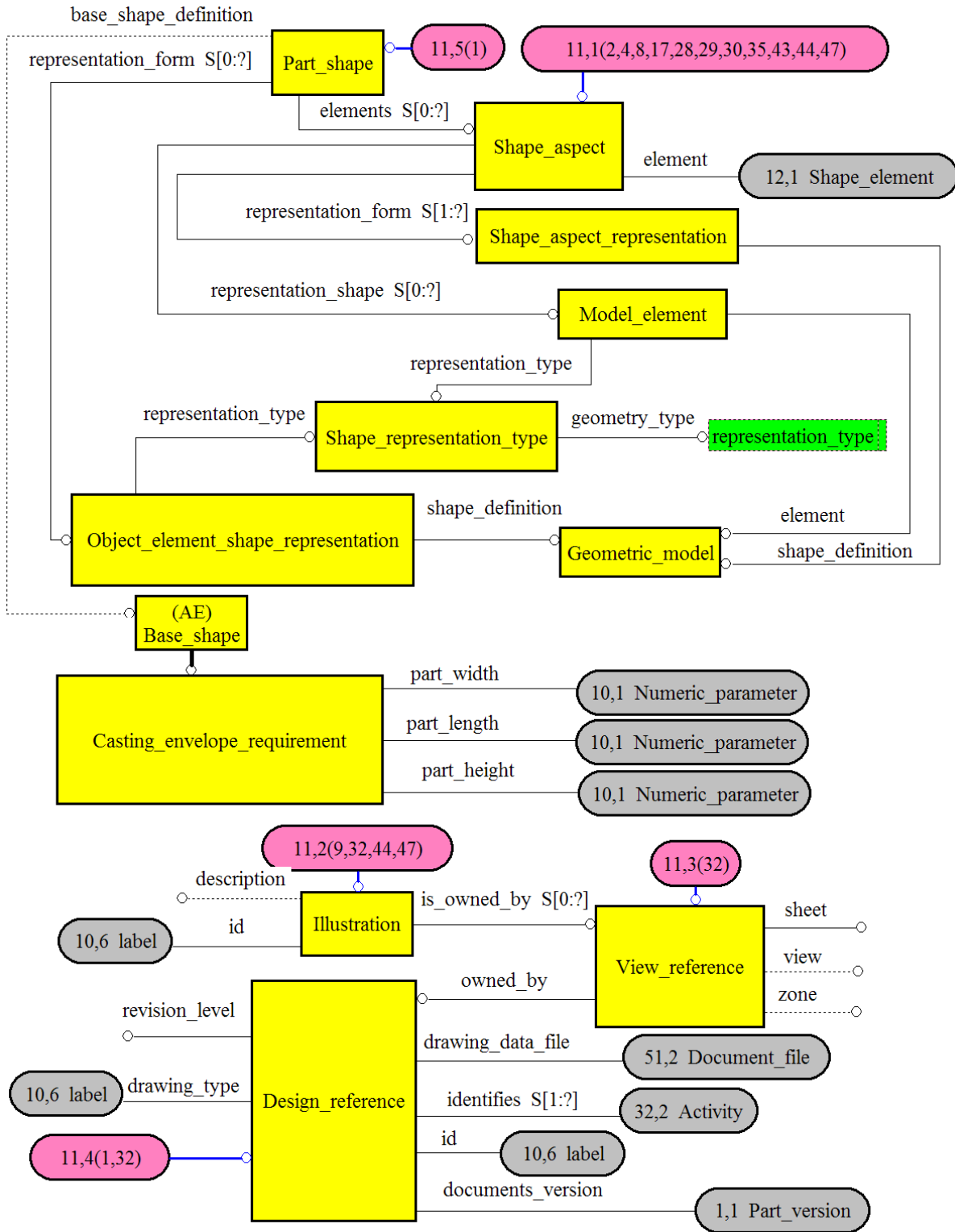


Figure G.11 — ARM diagram (11 of 53)

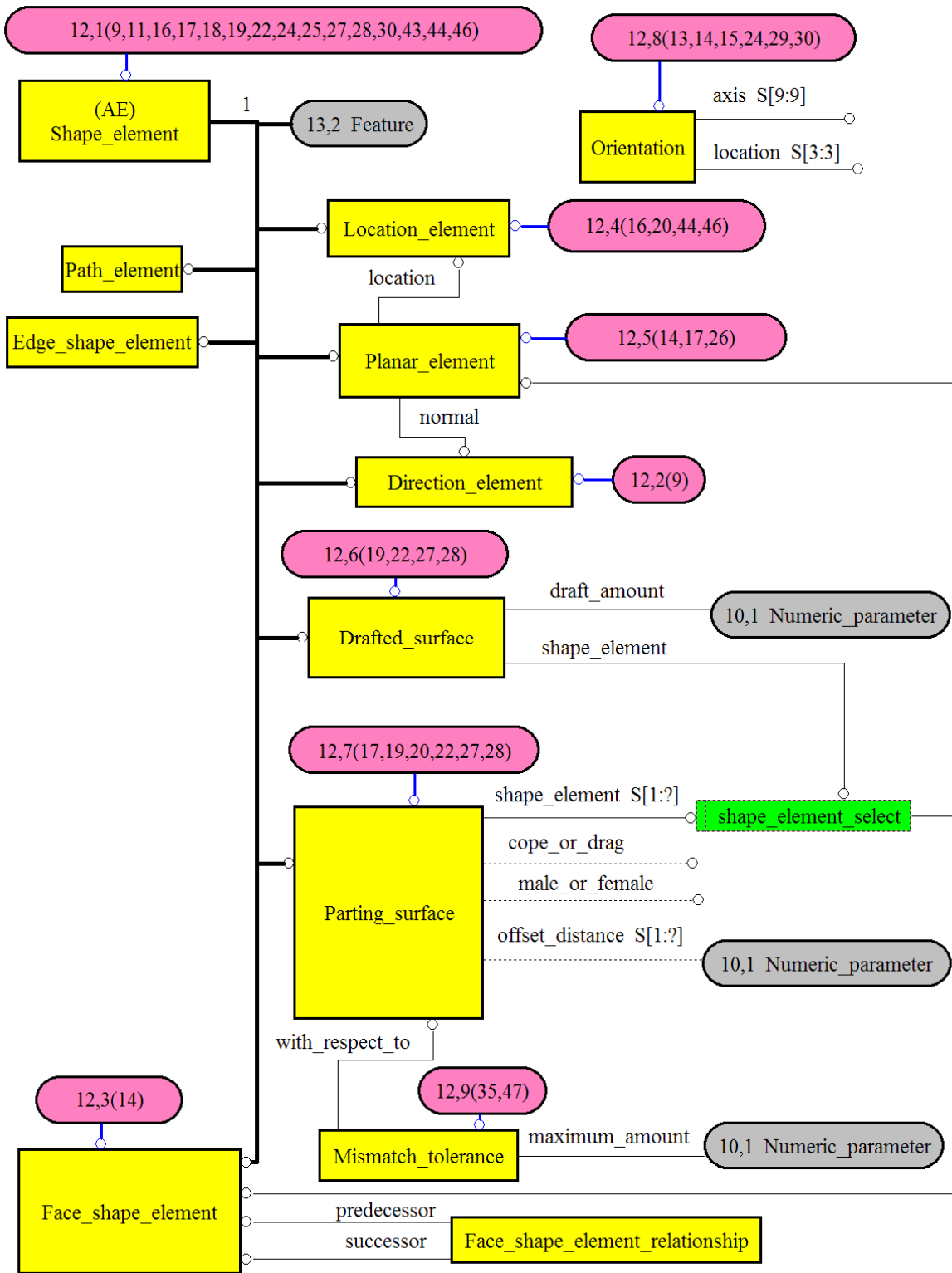


Figure G.12 — ARM diagram (12 of 53)

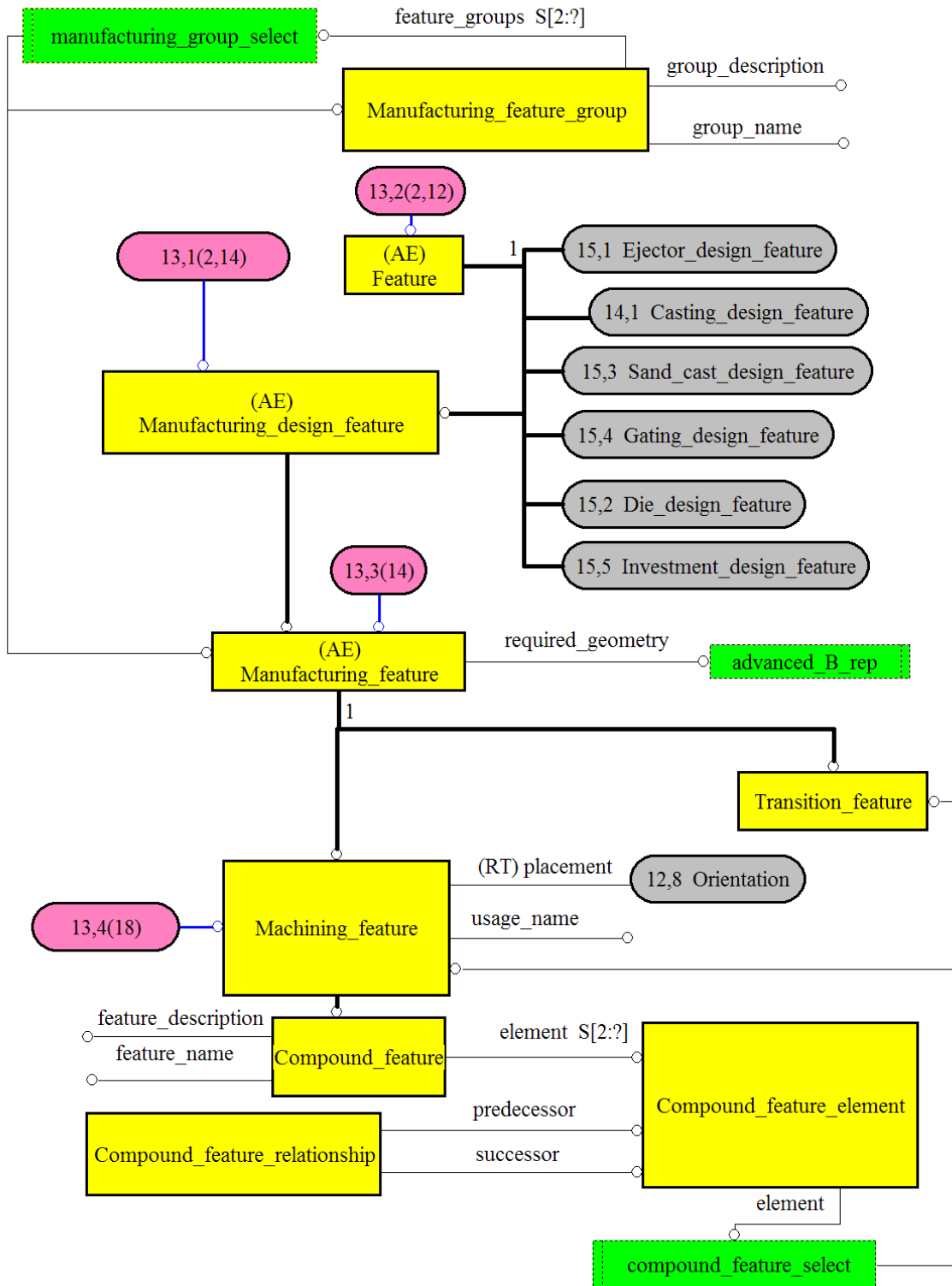


Figure G.13 — ARM diagram (13 of 53)

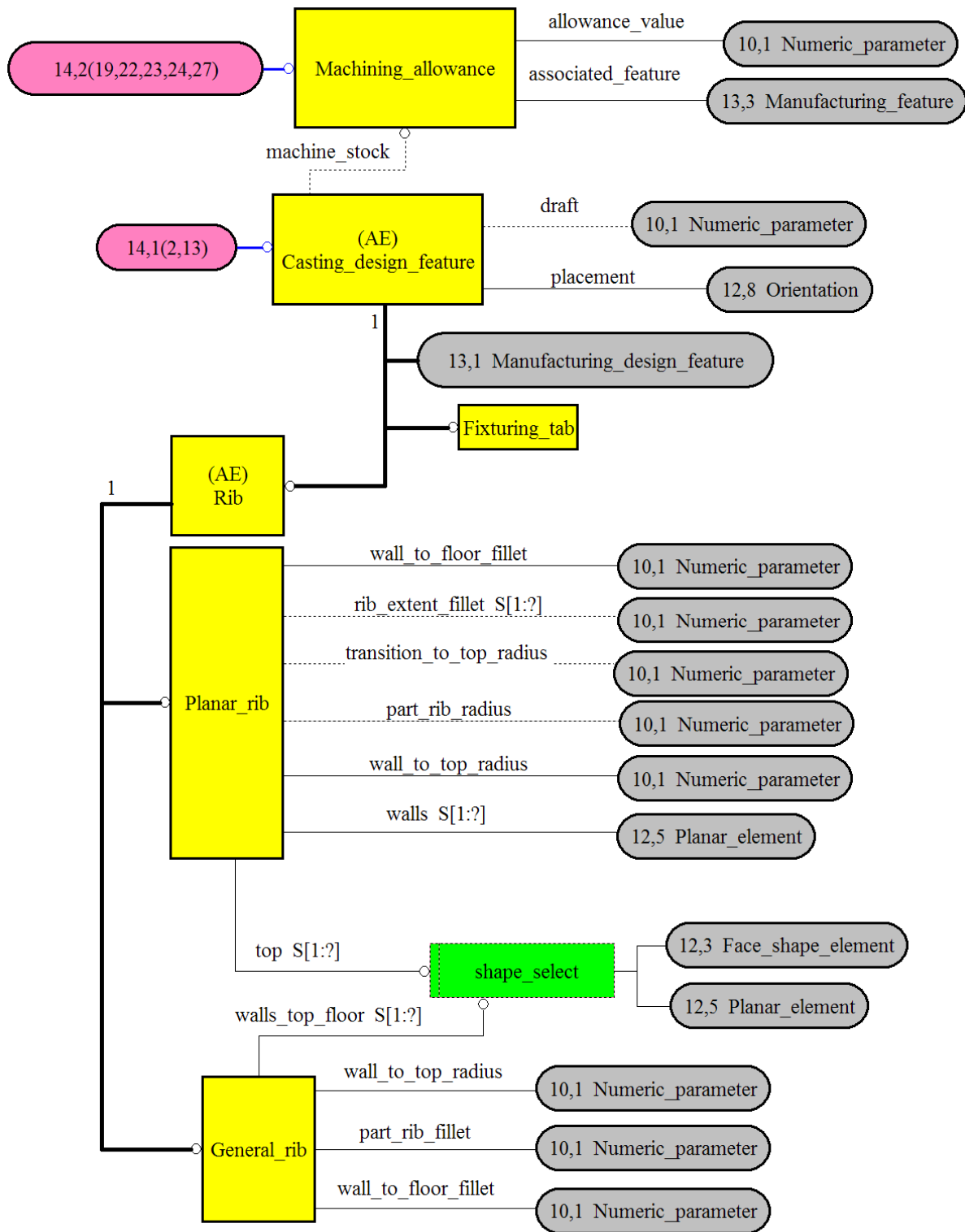


Figure G.14 — ARM diagram (14 of 53)

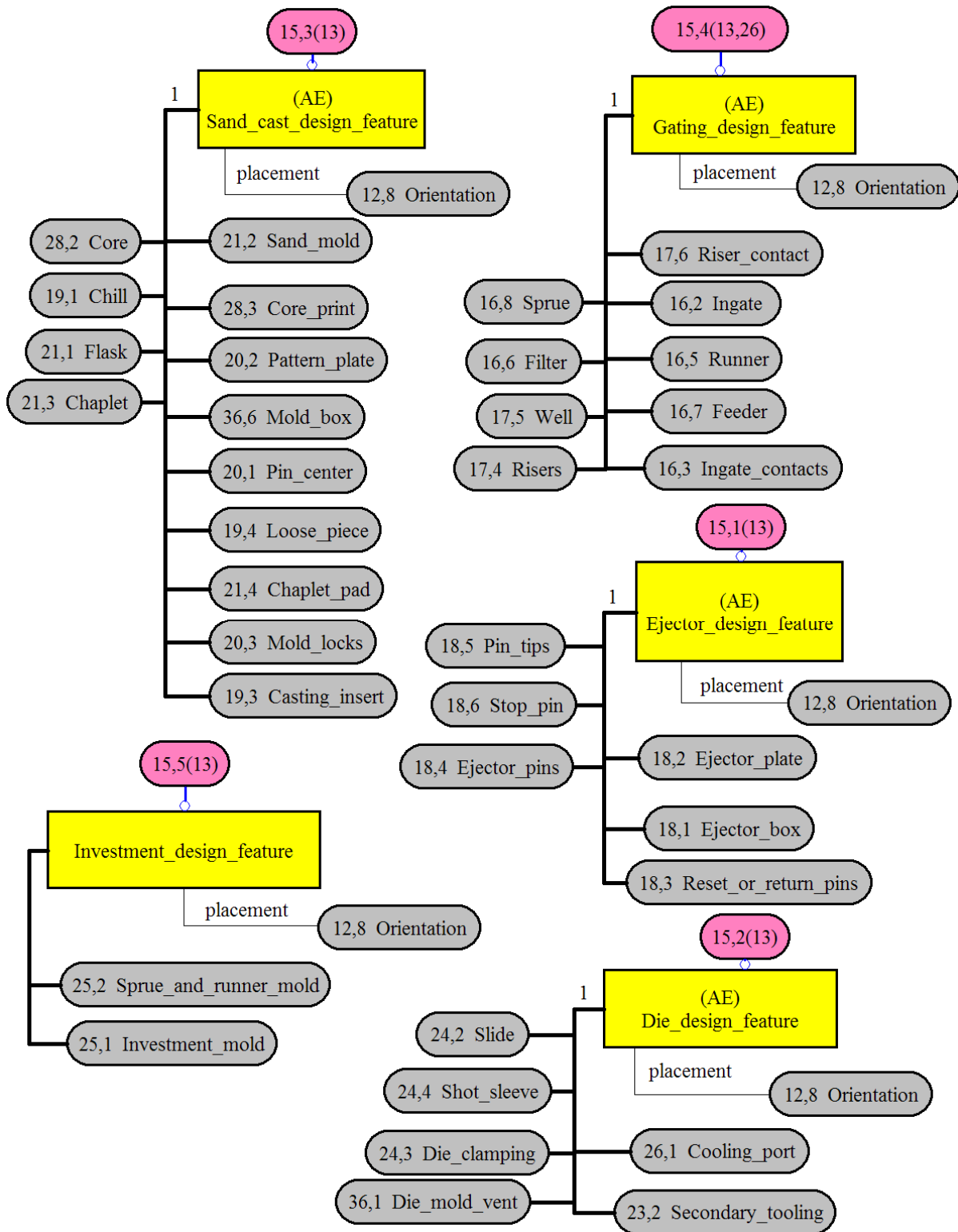


Figure G.15 — ARM diagram (15 of 53)

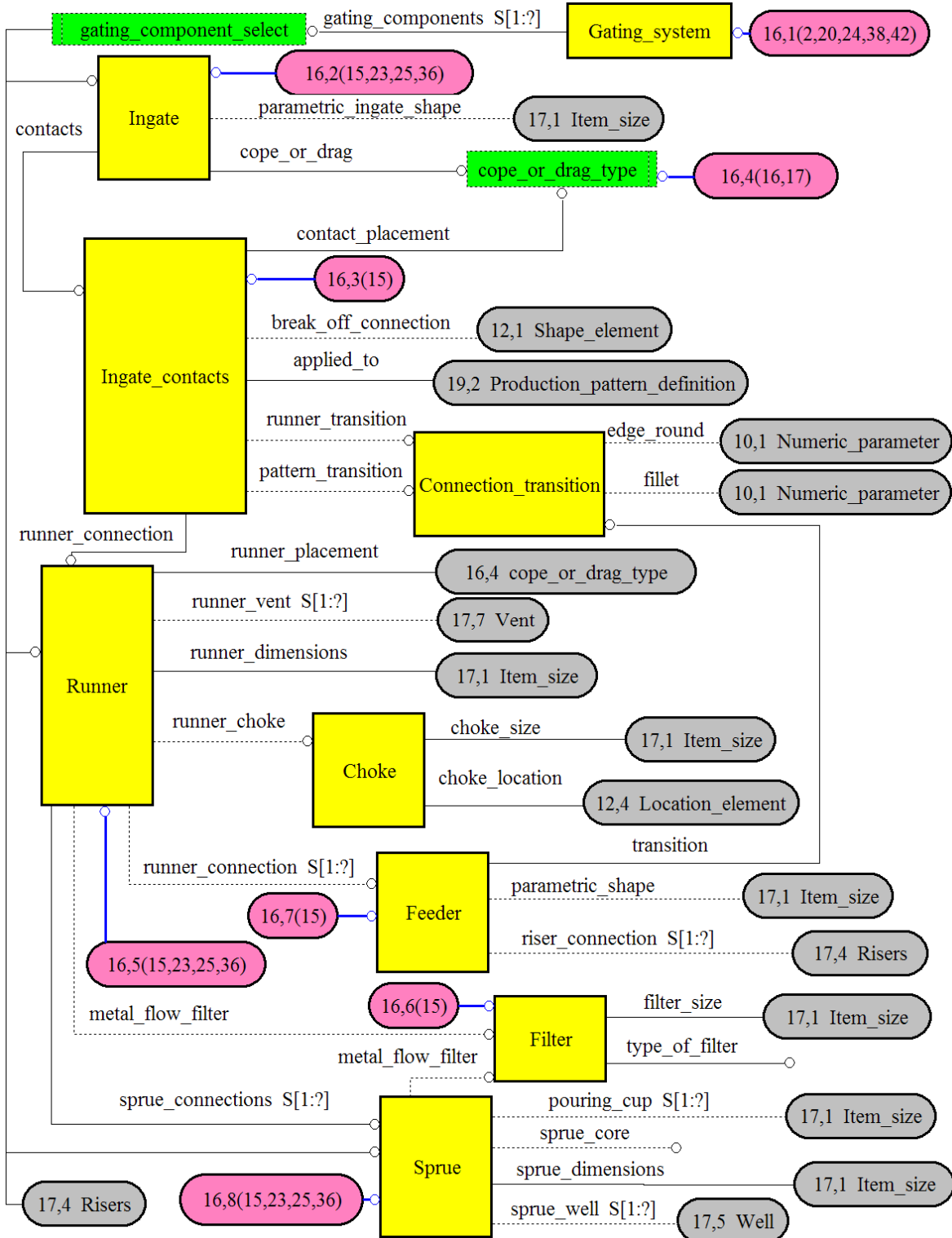


Figure G.16 — ARM diagram (16 of 53)



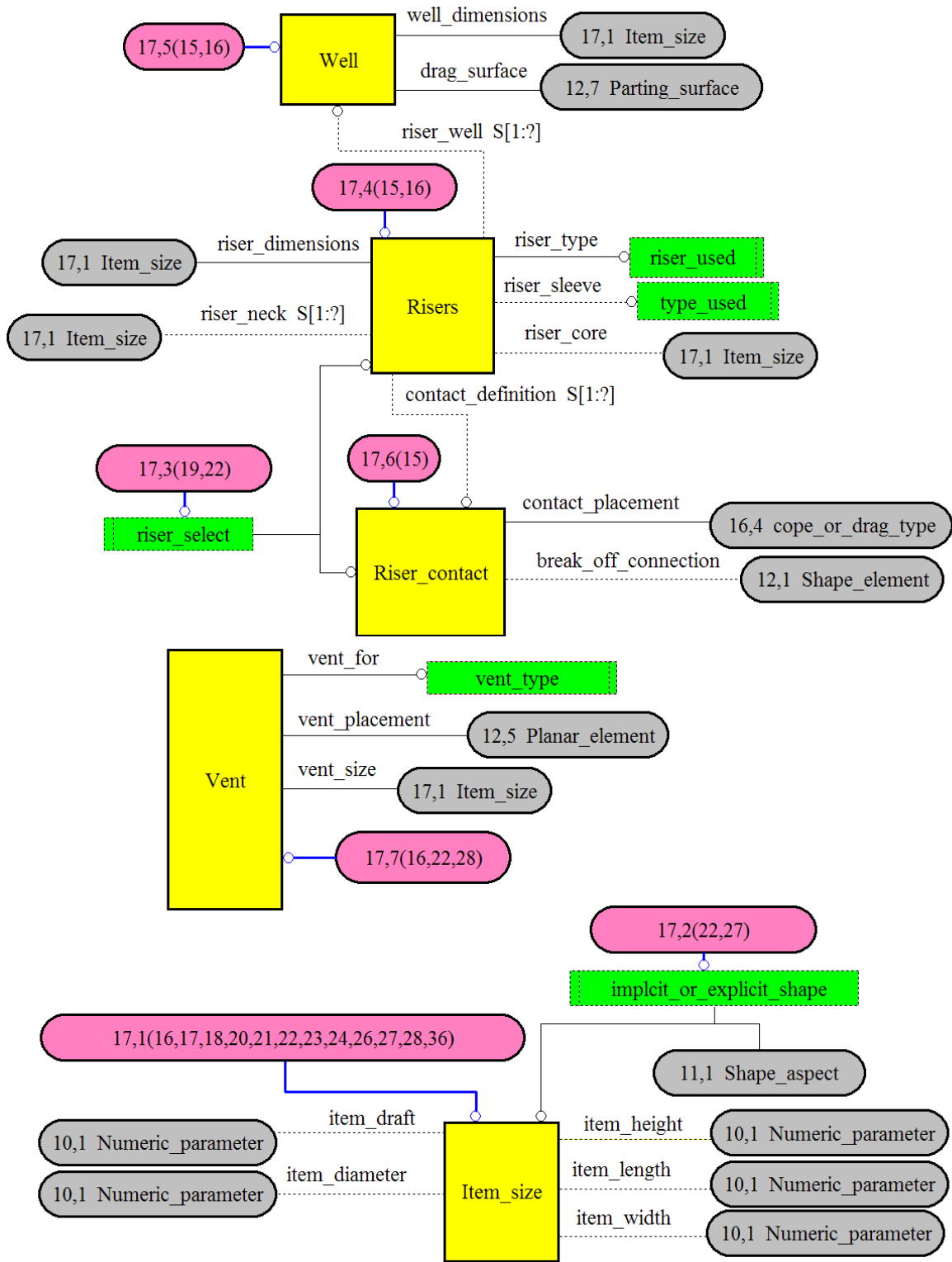


Figure G.17 — ARM diagram (17 of 53)

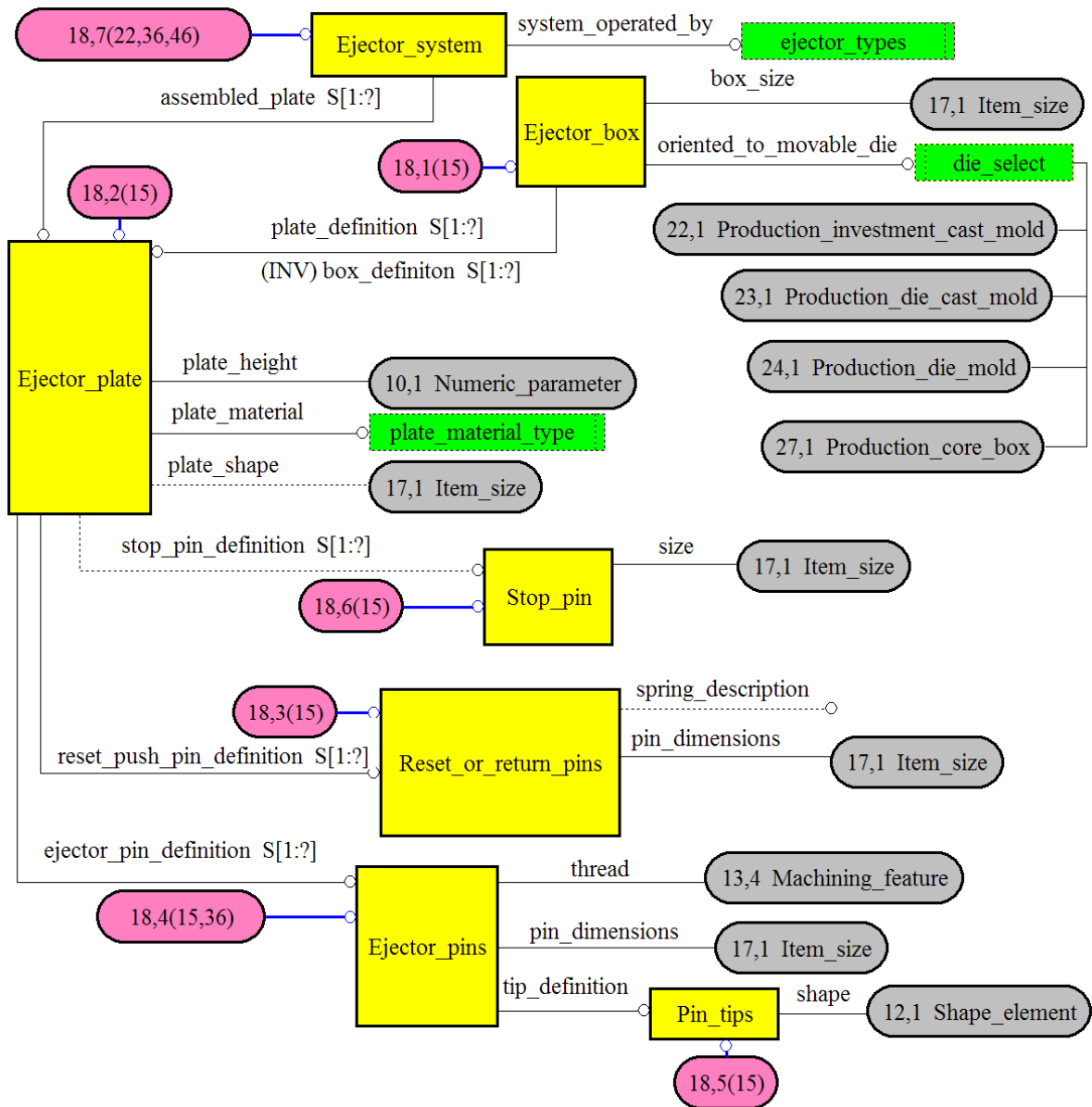


Figure G.18 — ARM diagram (18 of 53)

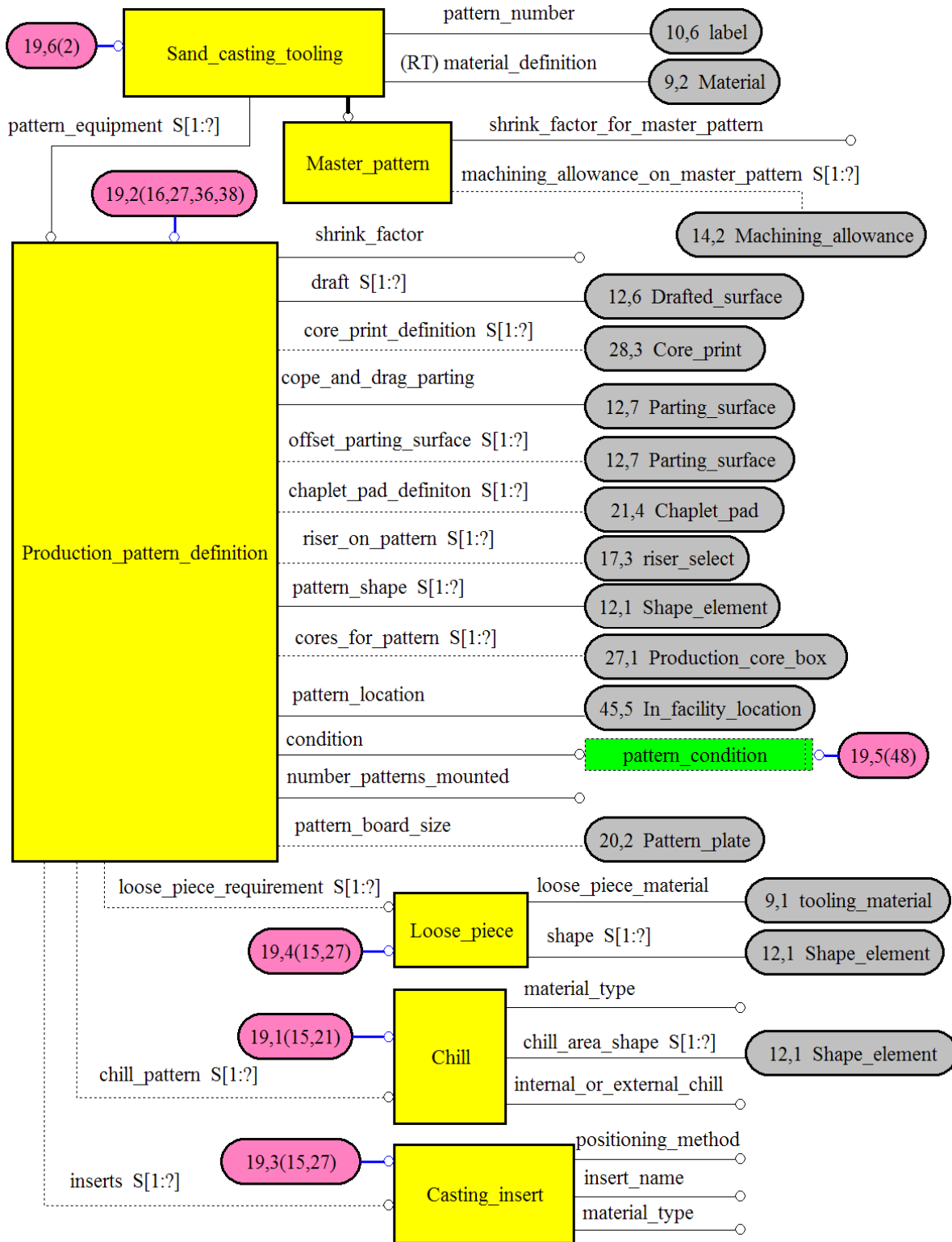
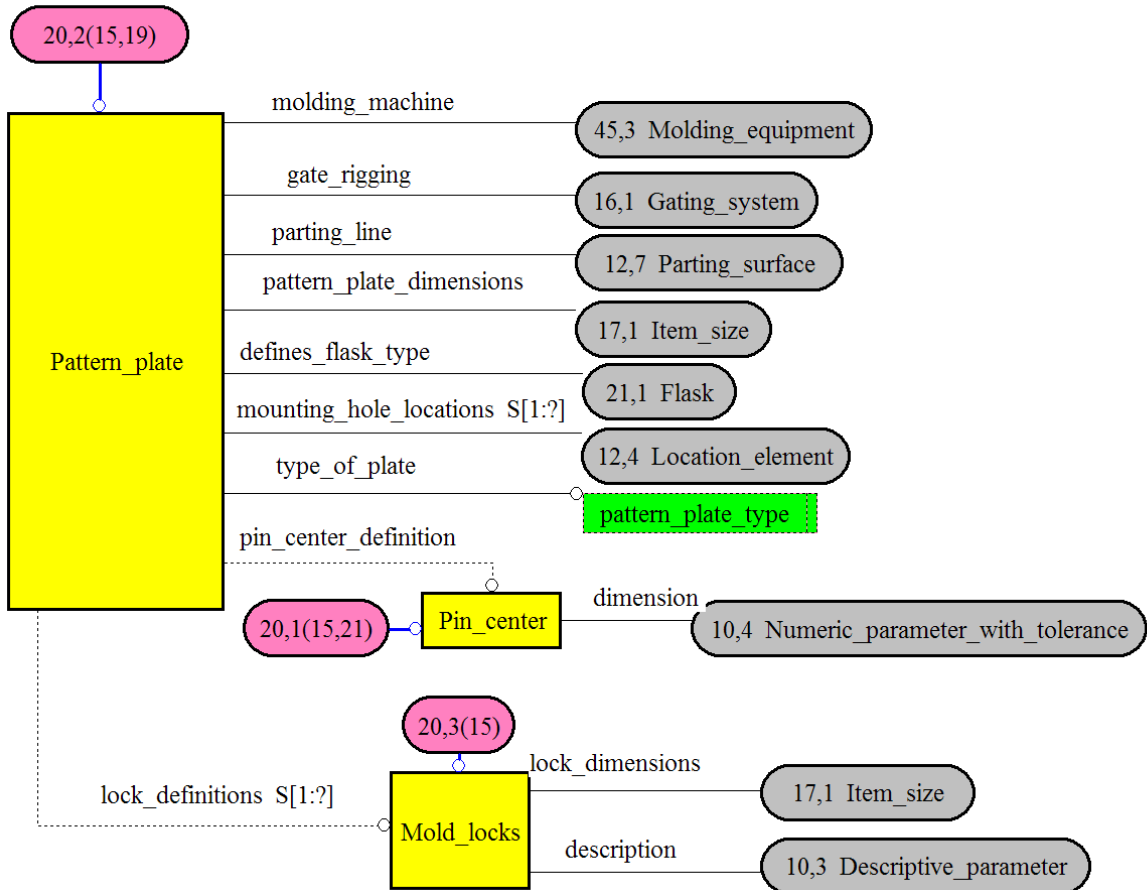


Figure G.19 — ARM diagram (19 of 53)



**Figure G.20 — ARM diagram (20 of 53)**

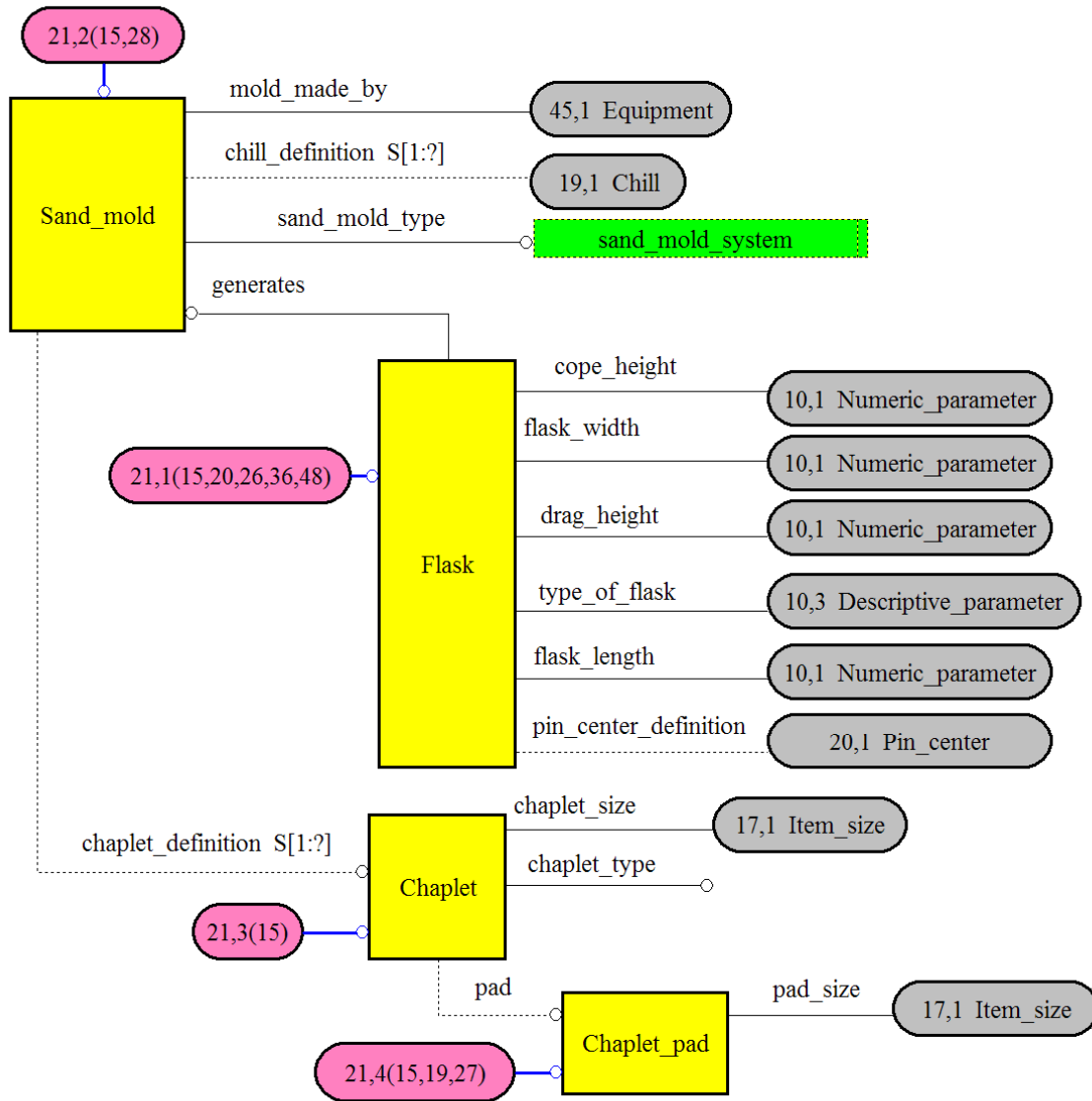
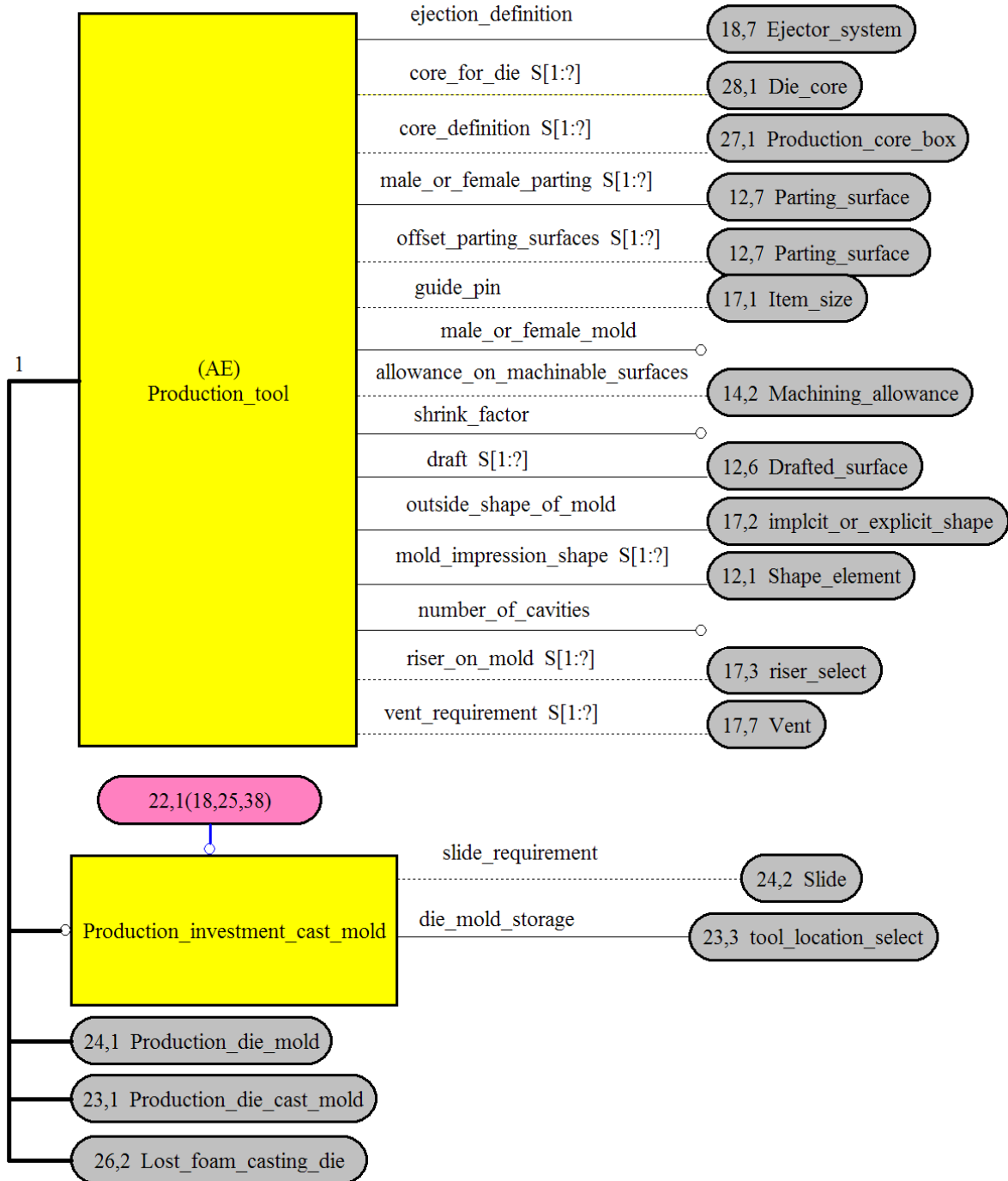
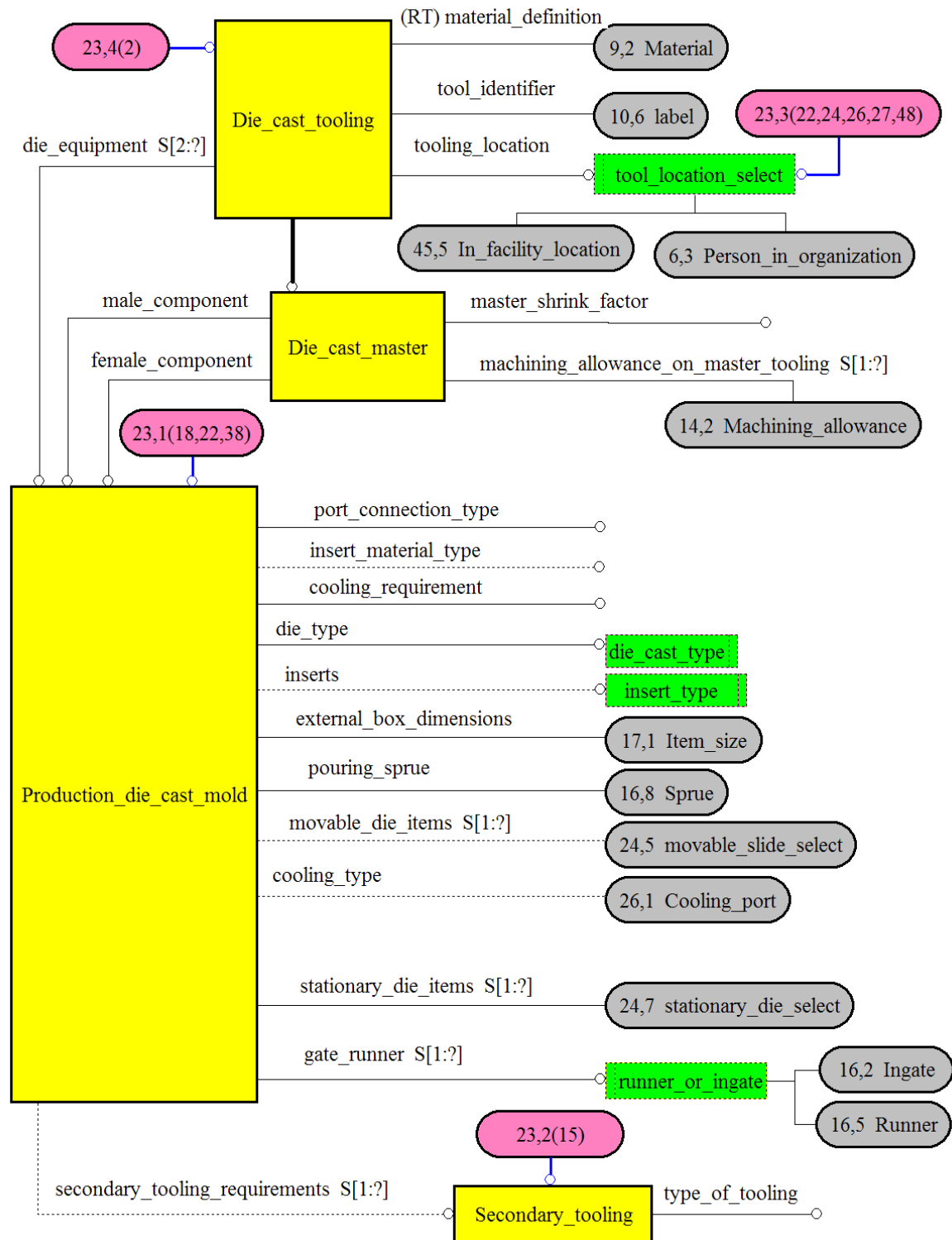


Figure G.21 — ARM diagram (21 of 53)



**Figure G.22 — ARM diagram (22 of 53)**



**Figure G.23 — ARM diagram (23 of 53)**

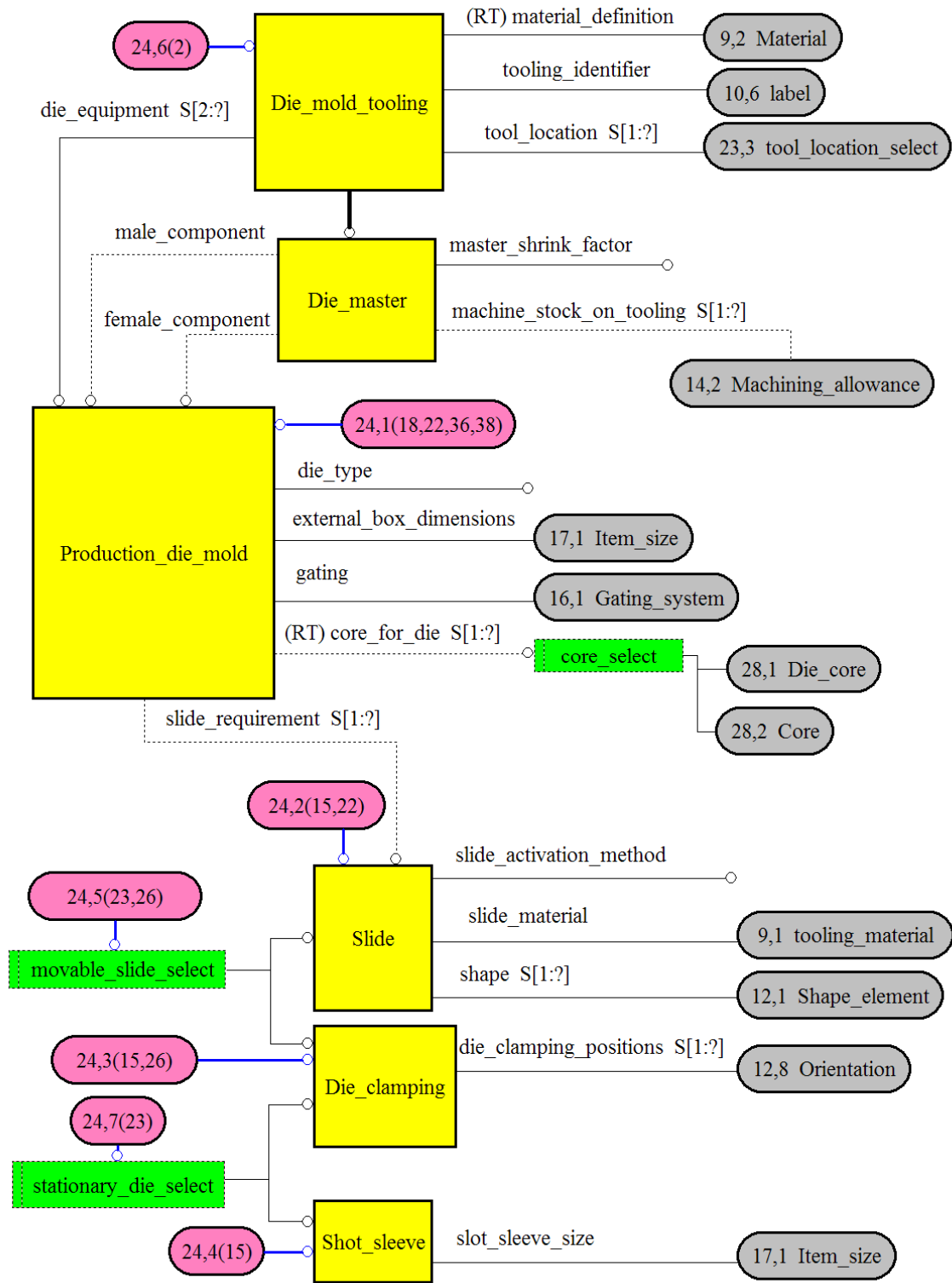
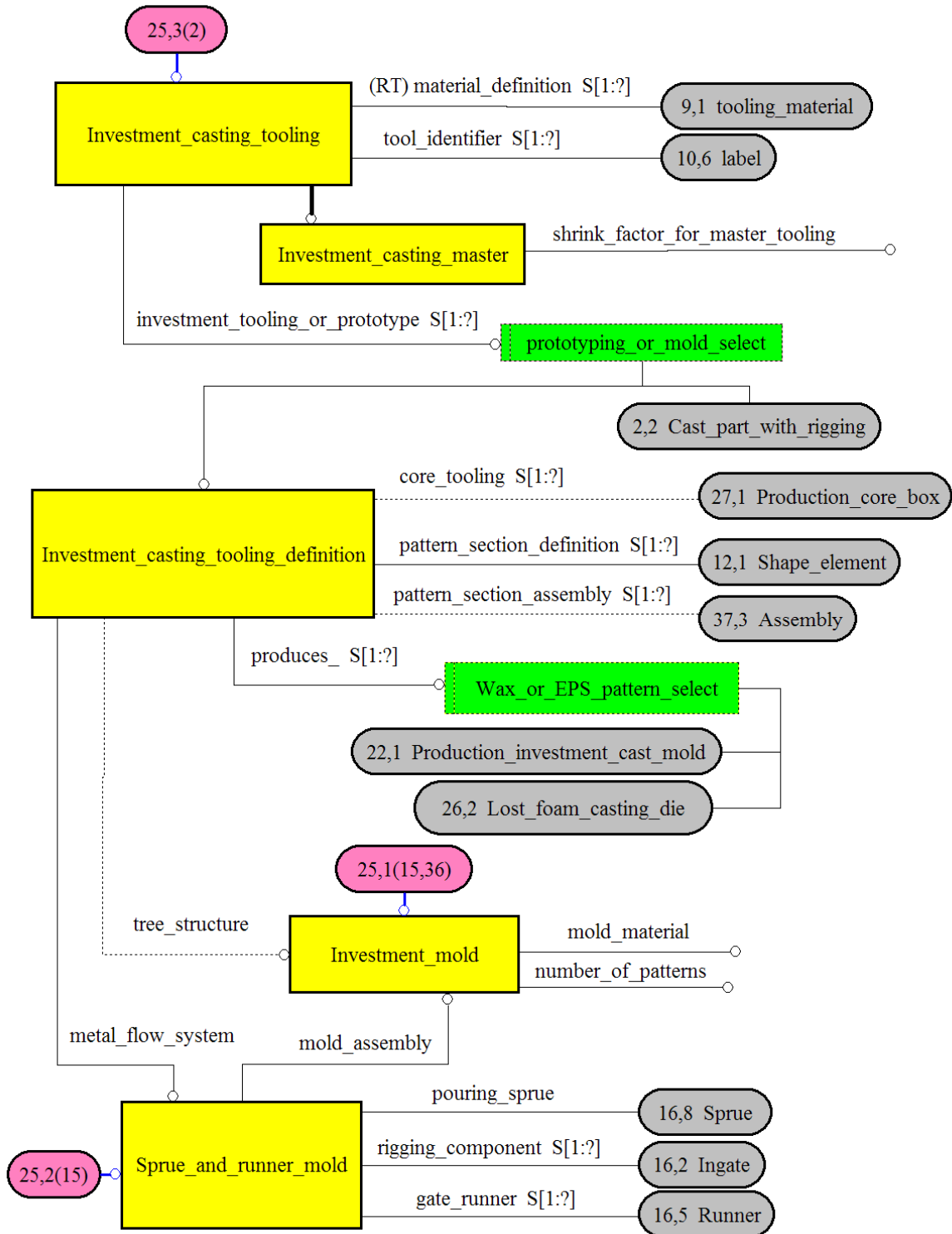
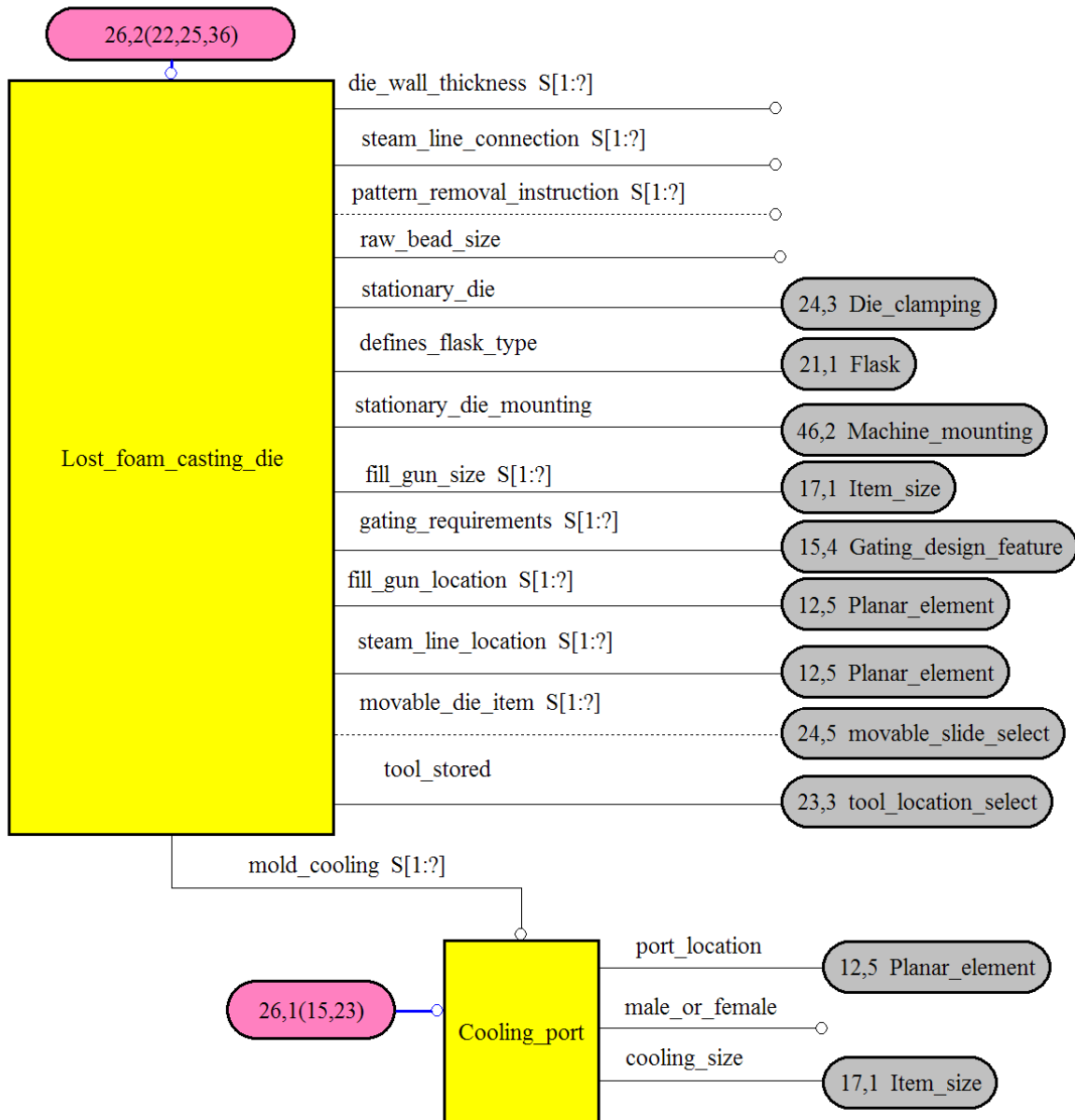


Figure G.24 — ARM diagram (24 of 53)

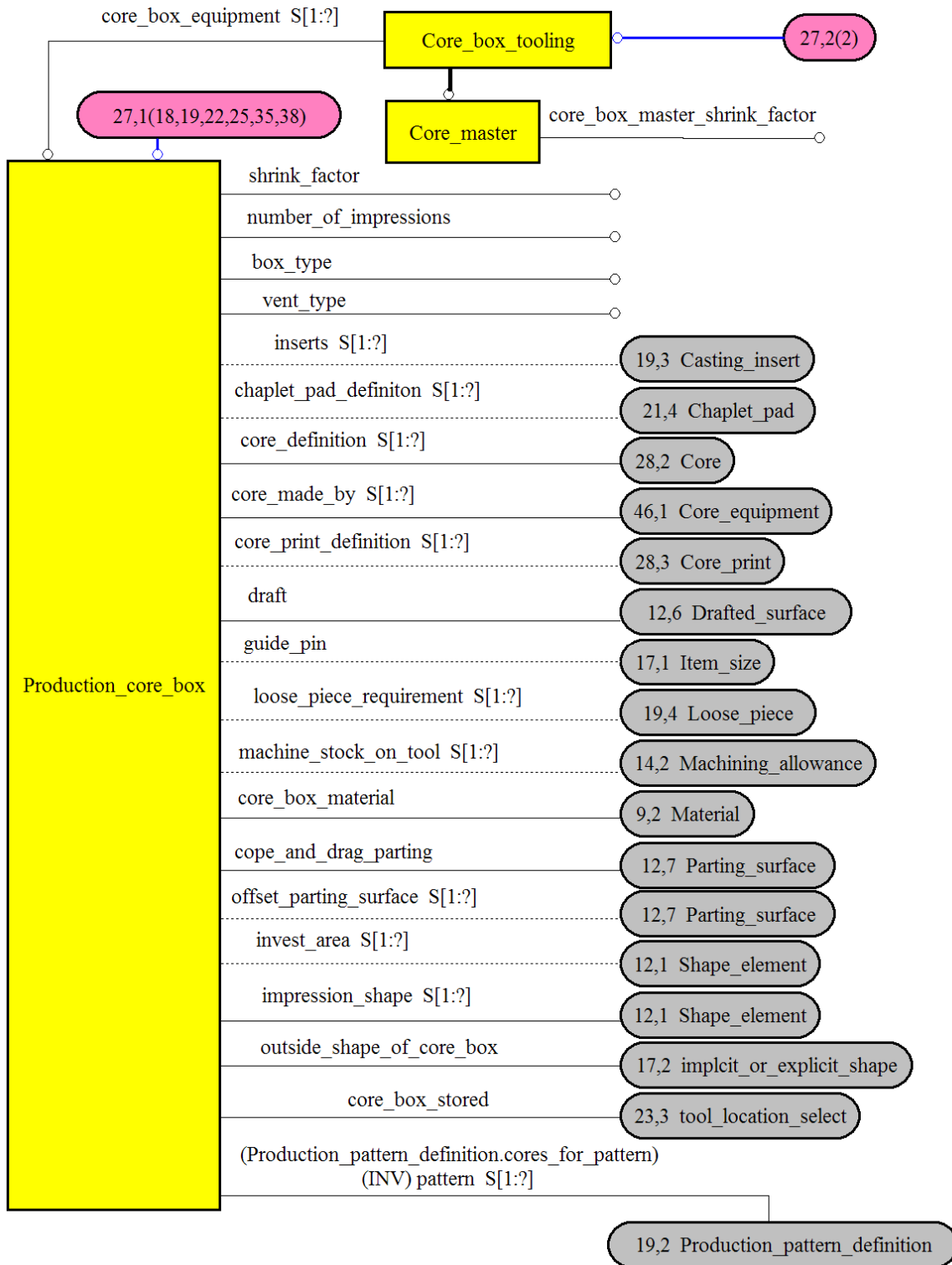




**Figure G.25 — ARM diagram (25 of 53)**



**Figure G.26 — ARM diagram (26 of 53)**



**Figure G.27 — ARM diagram (27 of 53)**

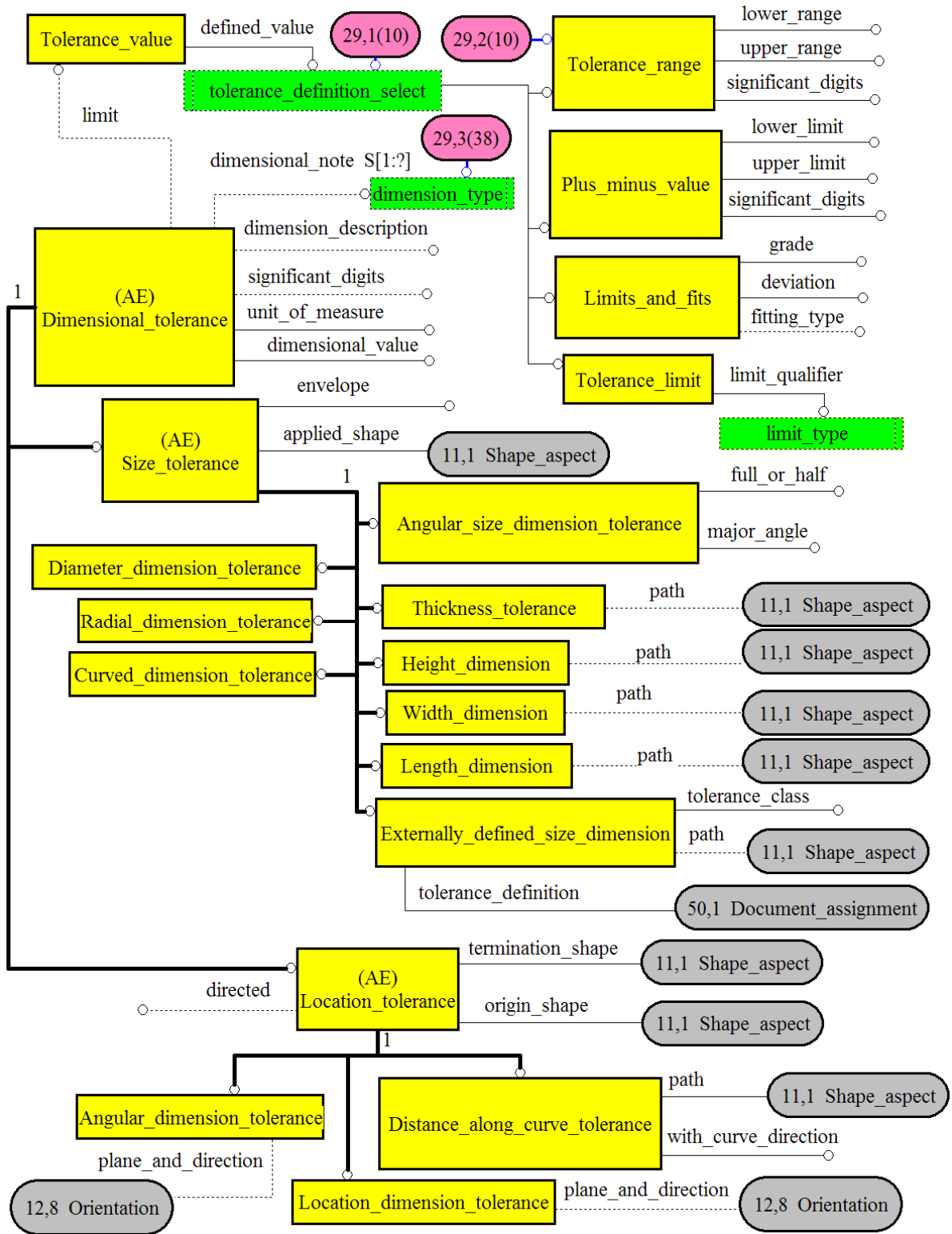


Figure G.28 — ARM diagram (28 of 53)

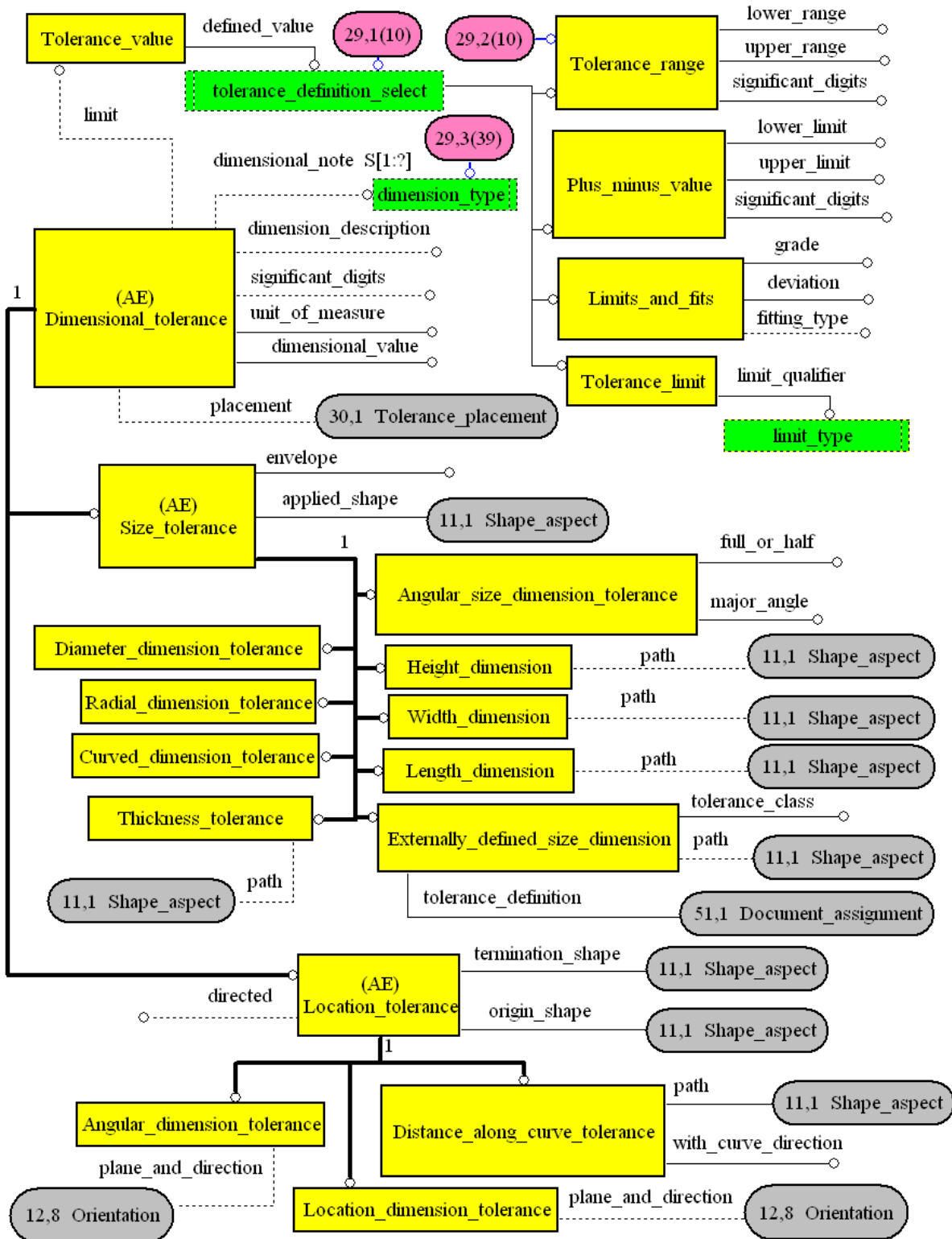
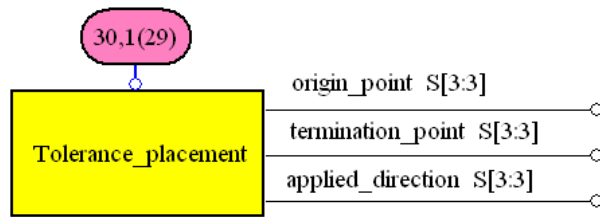


Figure G.29 — ARM diagram (29 of 53)



**Figure G.30 — ARM diagram (30 of 53)**

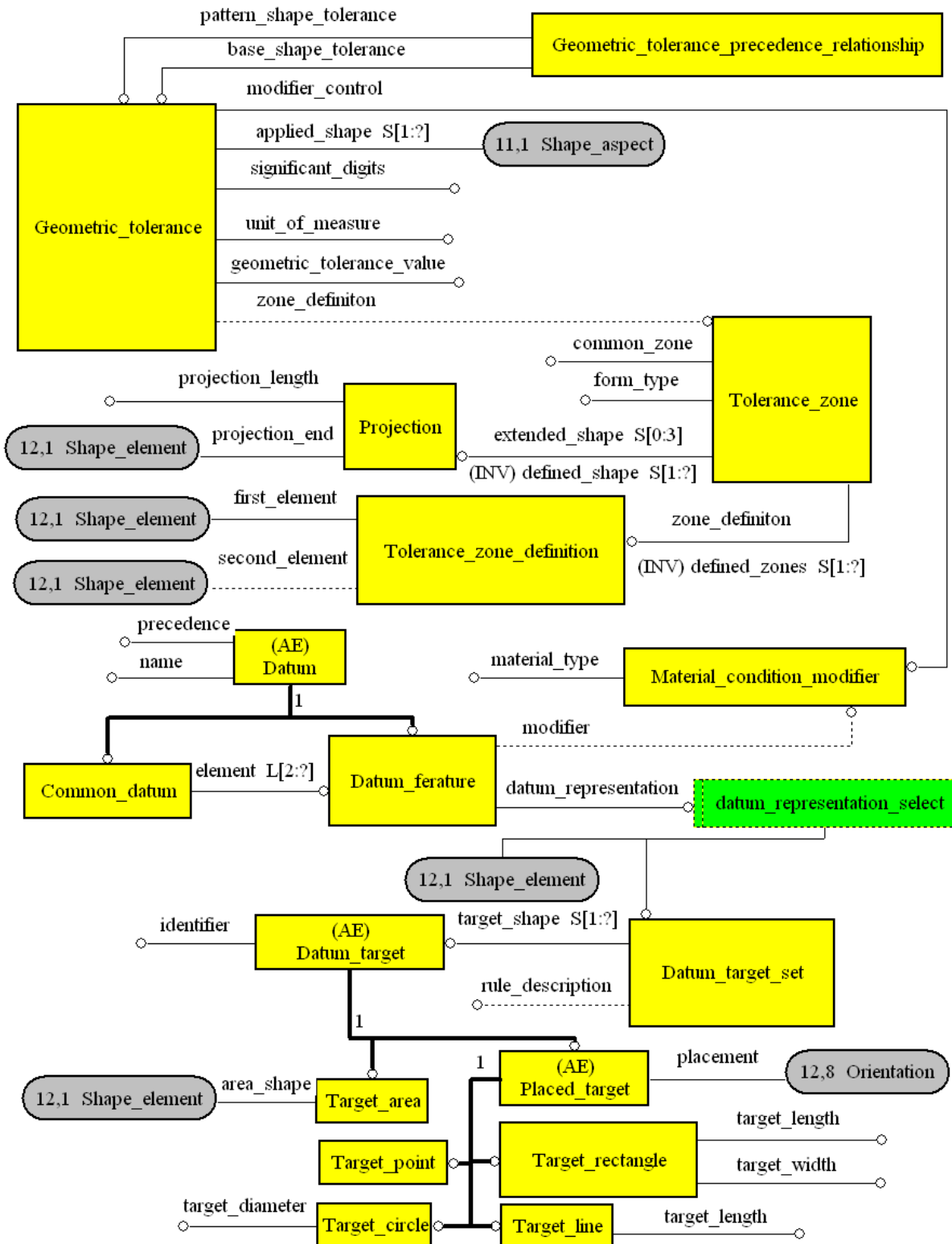


Figure G.31 — ARM diagram (31 of 53)

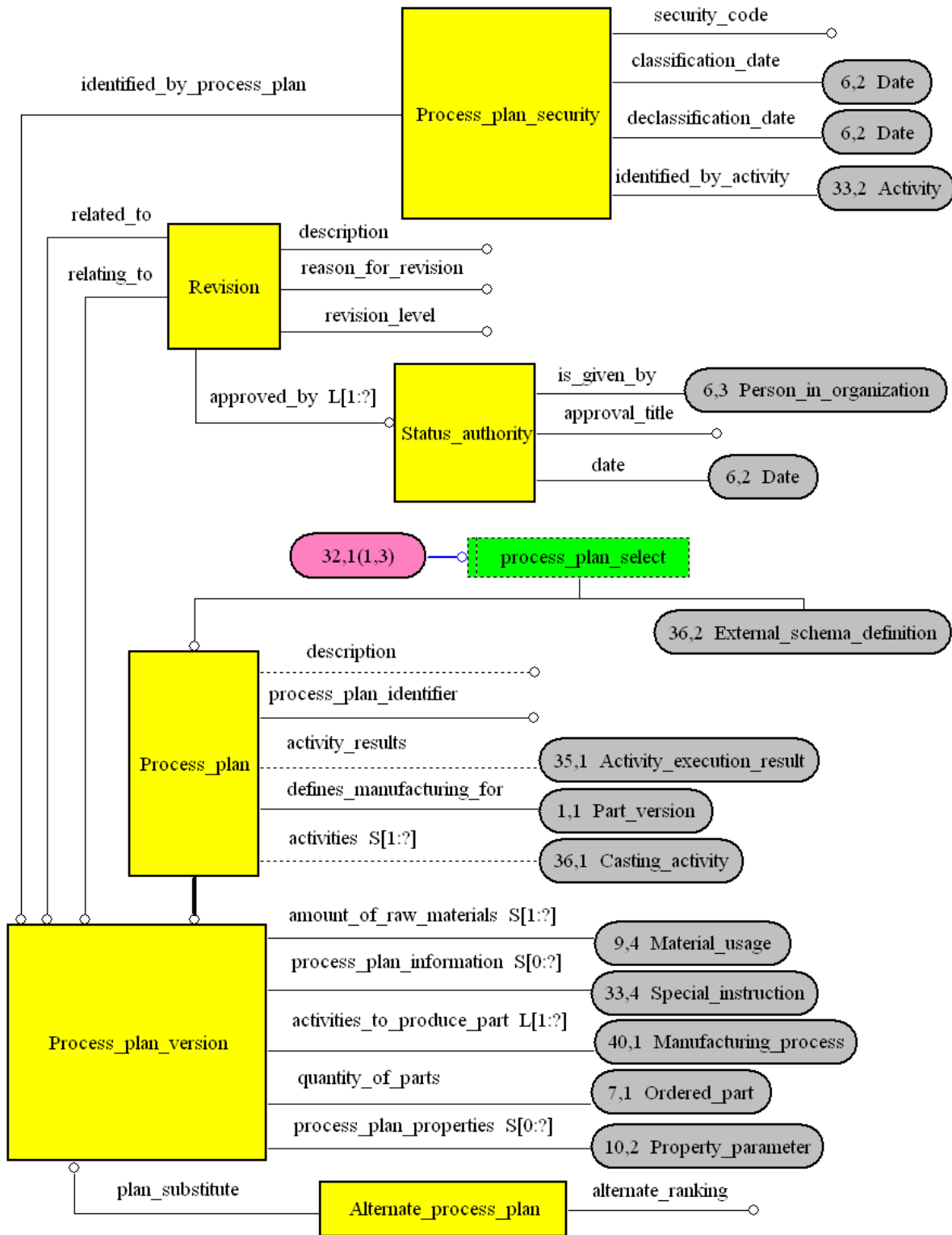


Figure G.32 — ARM diagram (32 of 53)



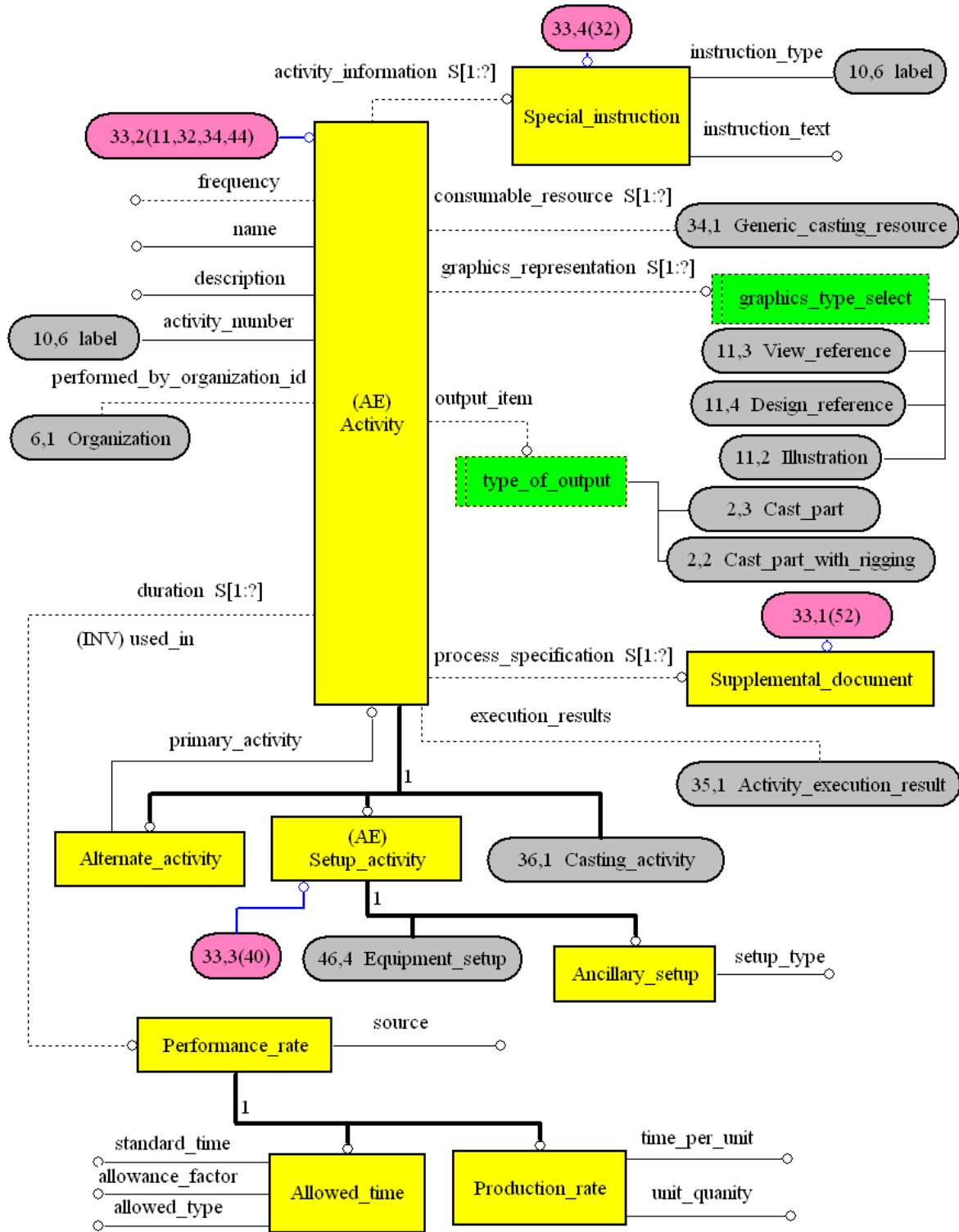


Figure G.33 — ARM diagram (33 of 53)

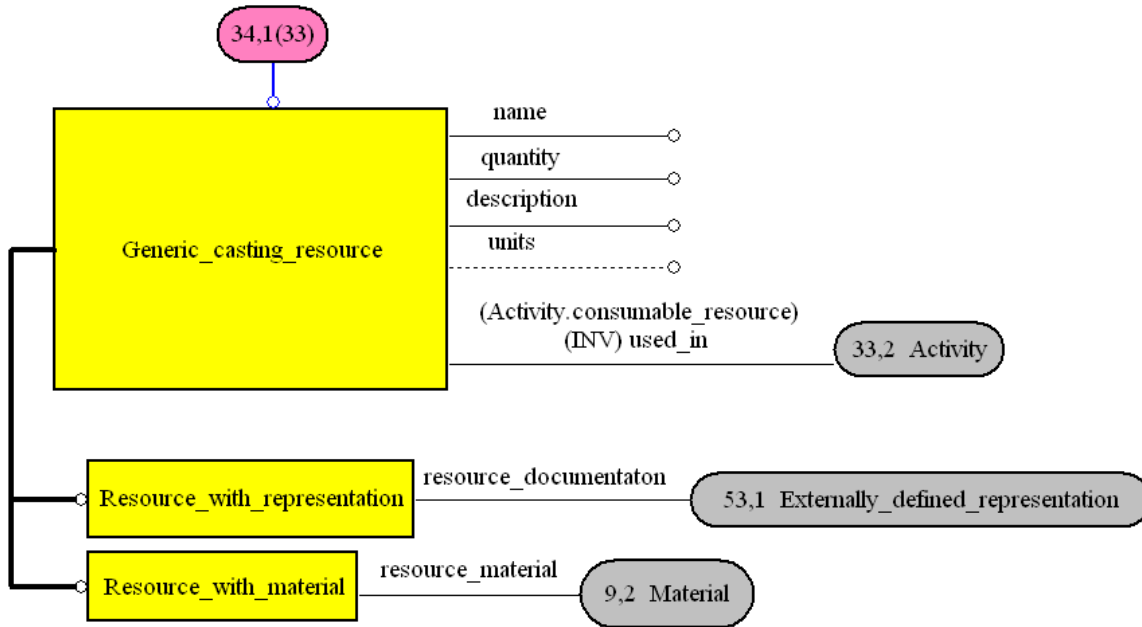
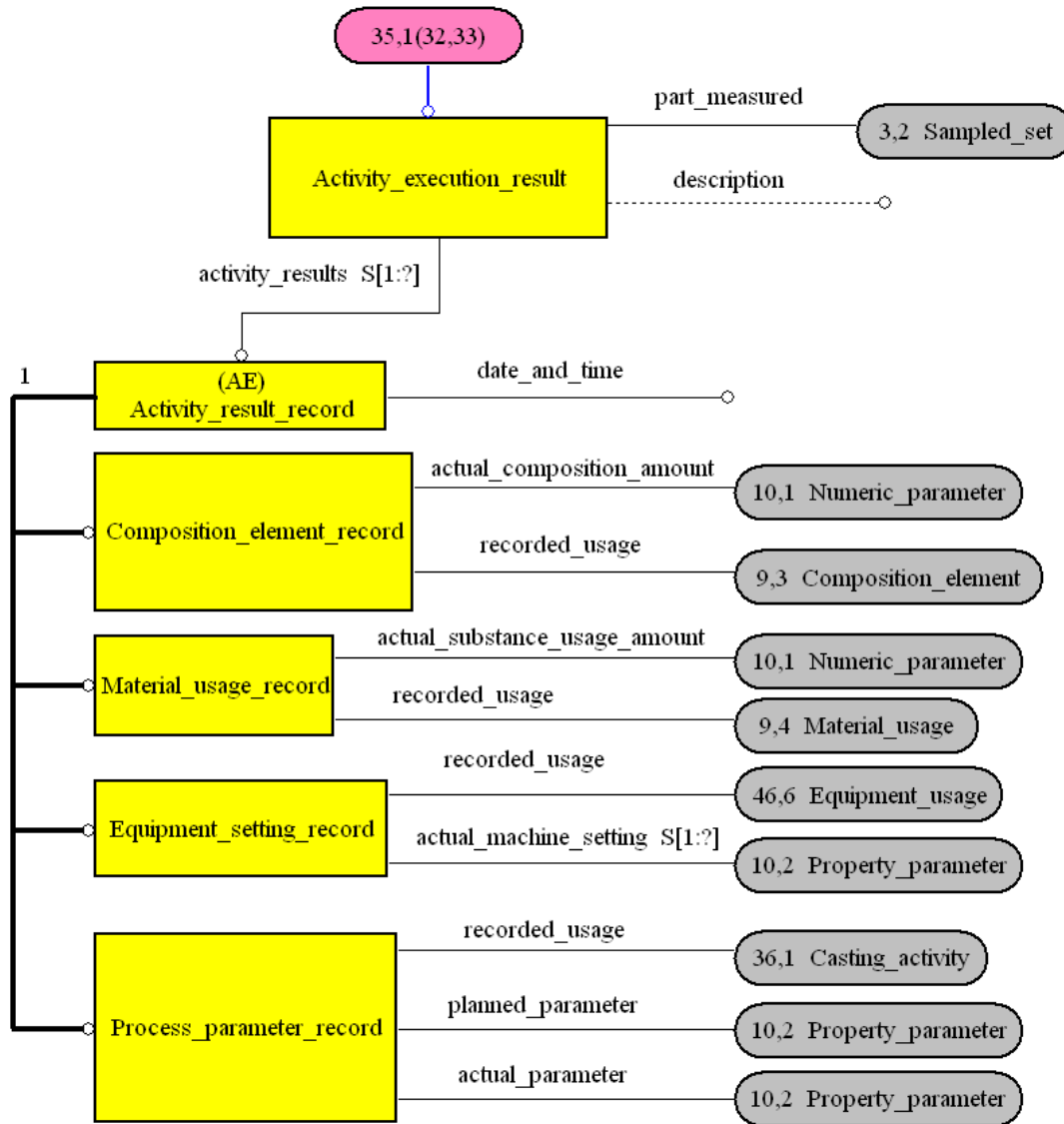


Figure G.34 — ARM diagram (34 of 53)



**Figure G.35 — ARM diagram (35 of 53)**

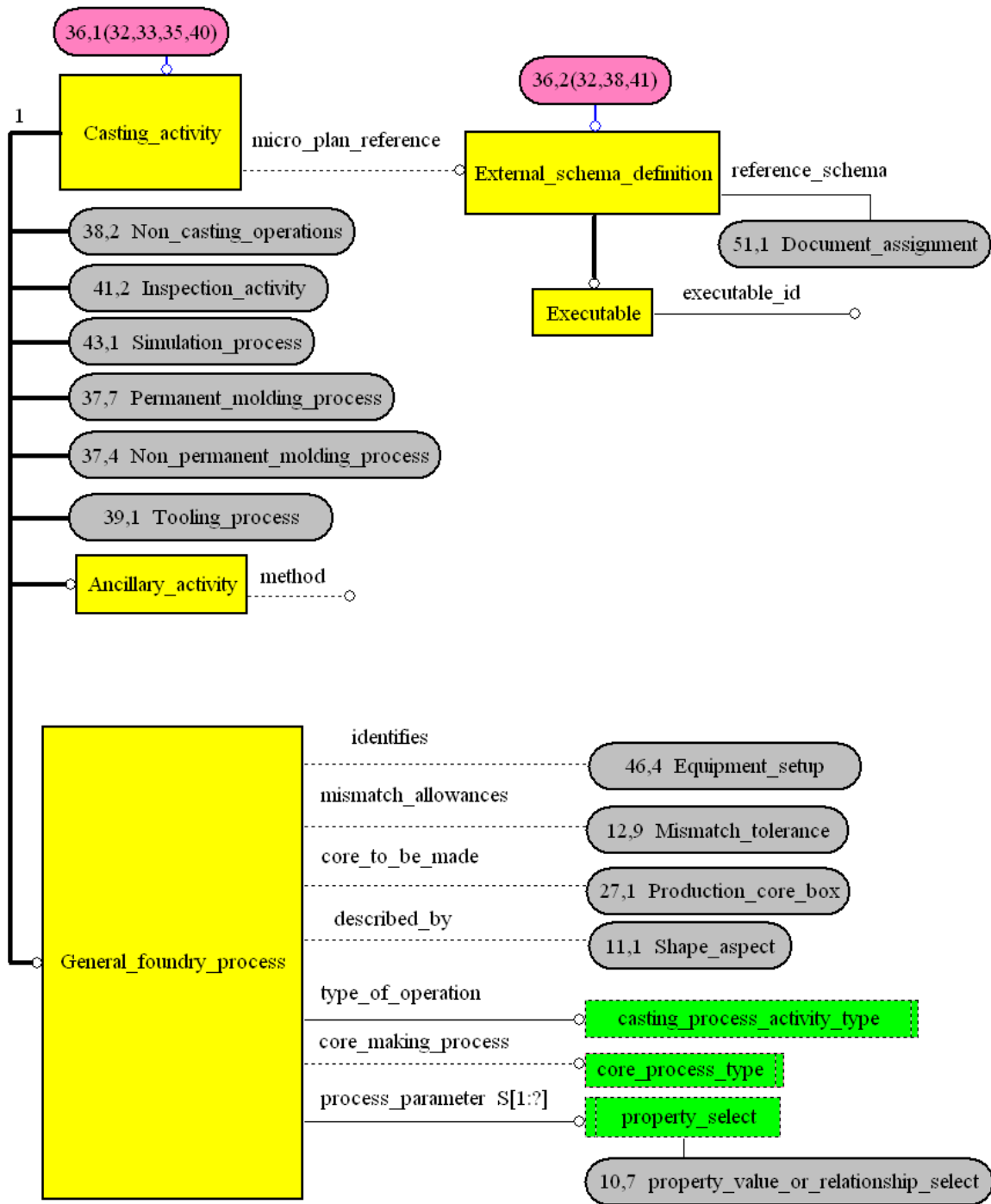


Figure G.36 — ARM diagram (36 of 53)

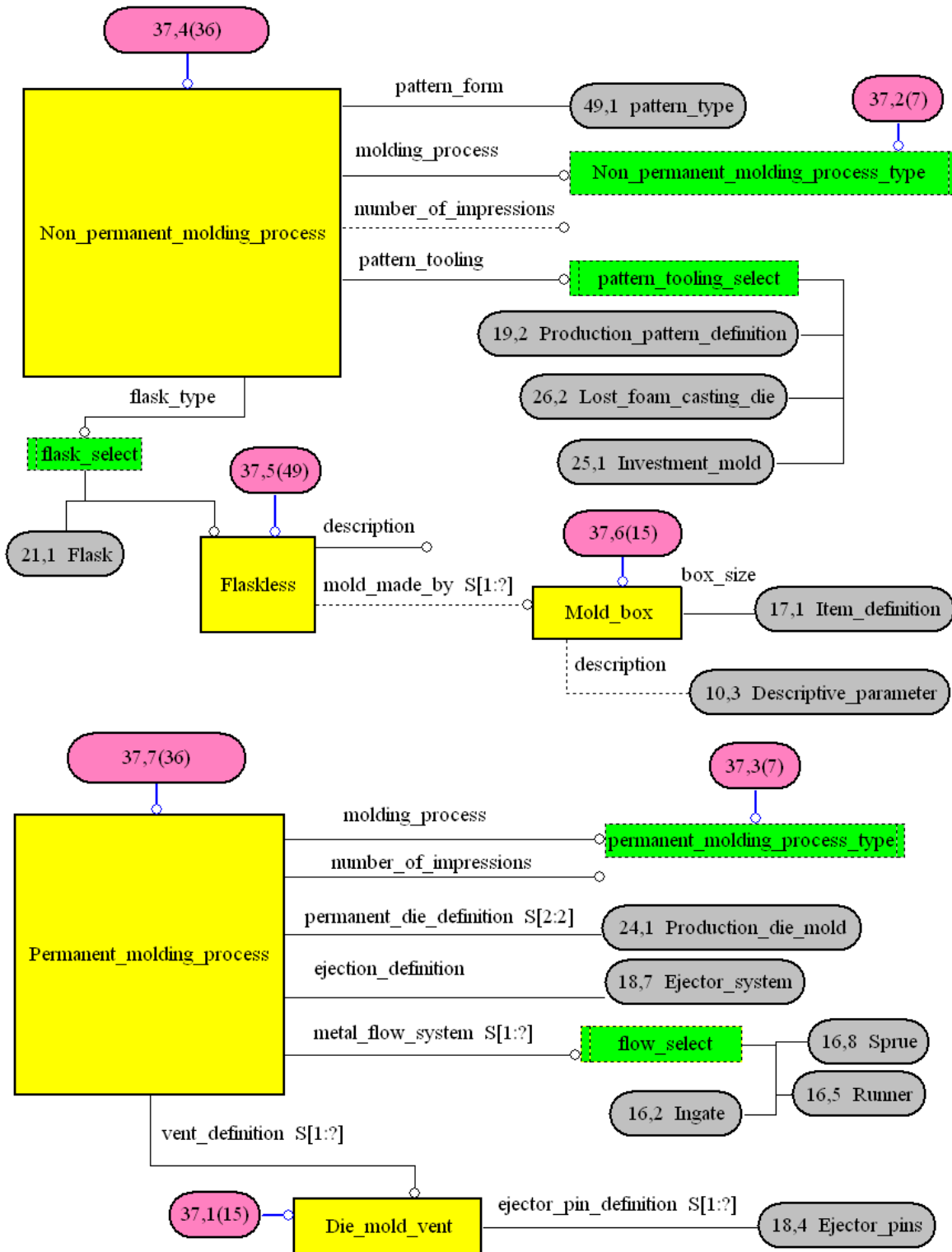


Figure G.37 — ARM diagram (37 of 53)

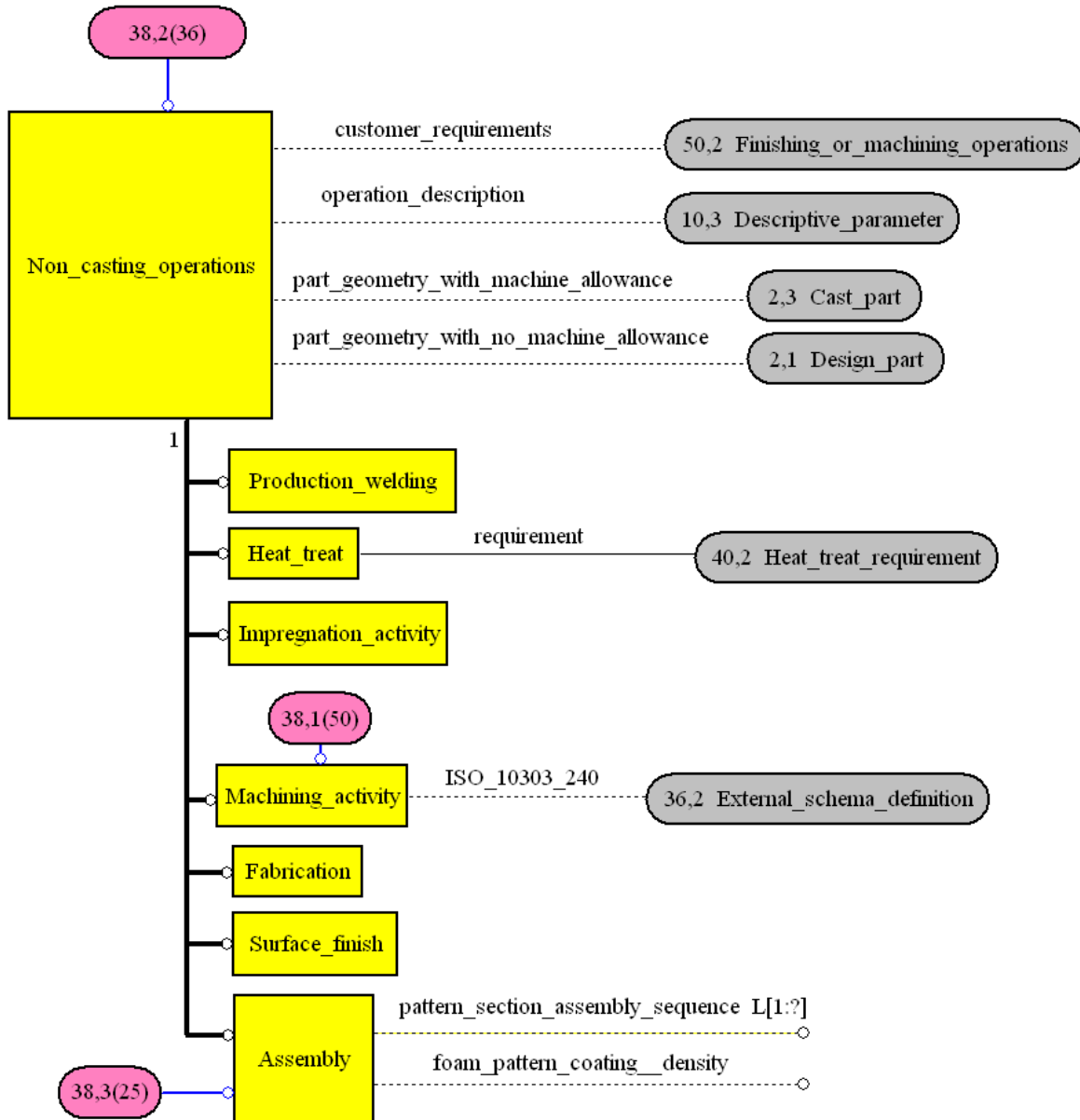


Figure G.38 — ARM diagram (38 of 53)

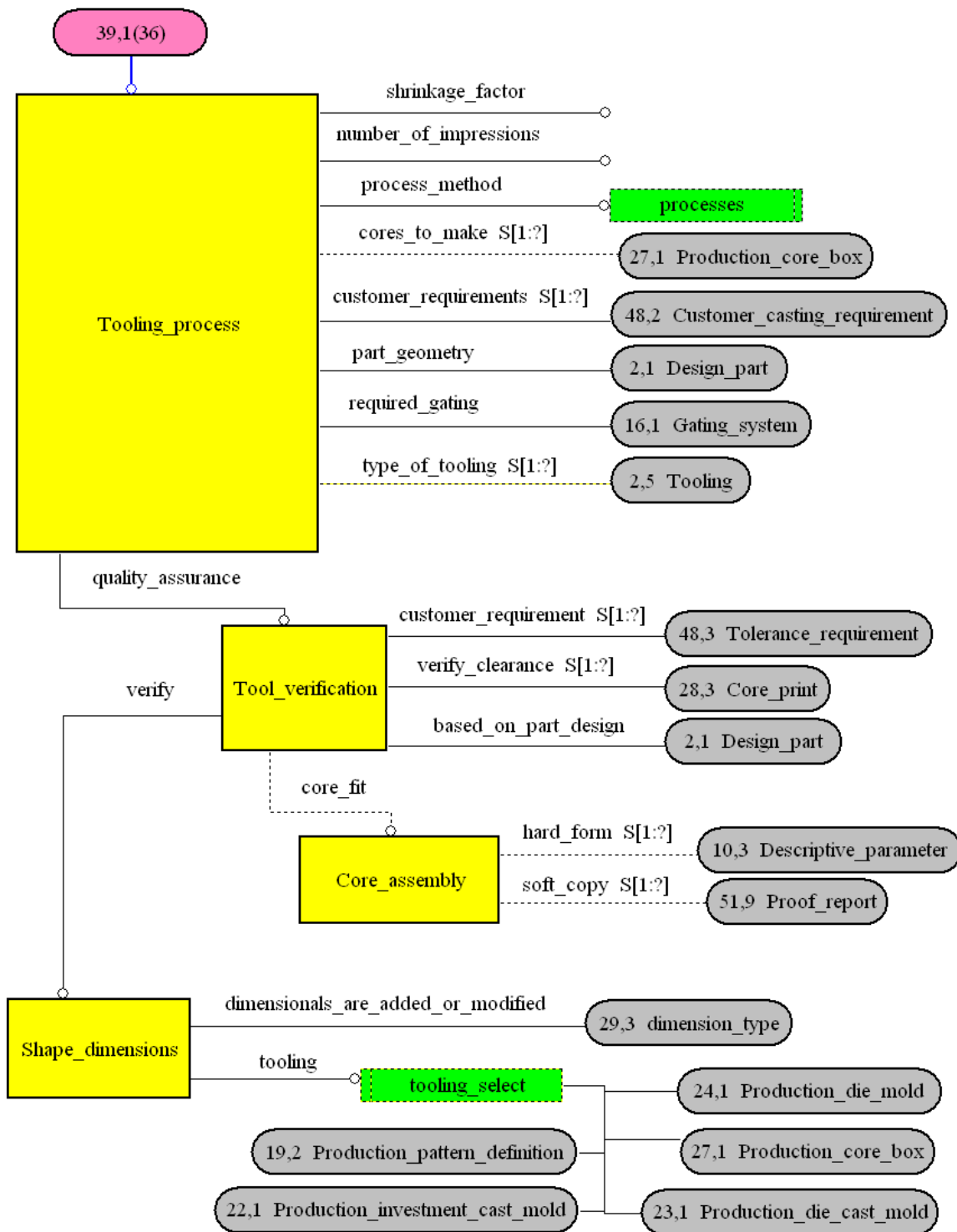


Figure G.39 — ARM diagram (39 of 53)

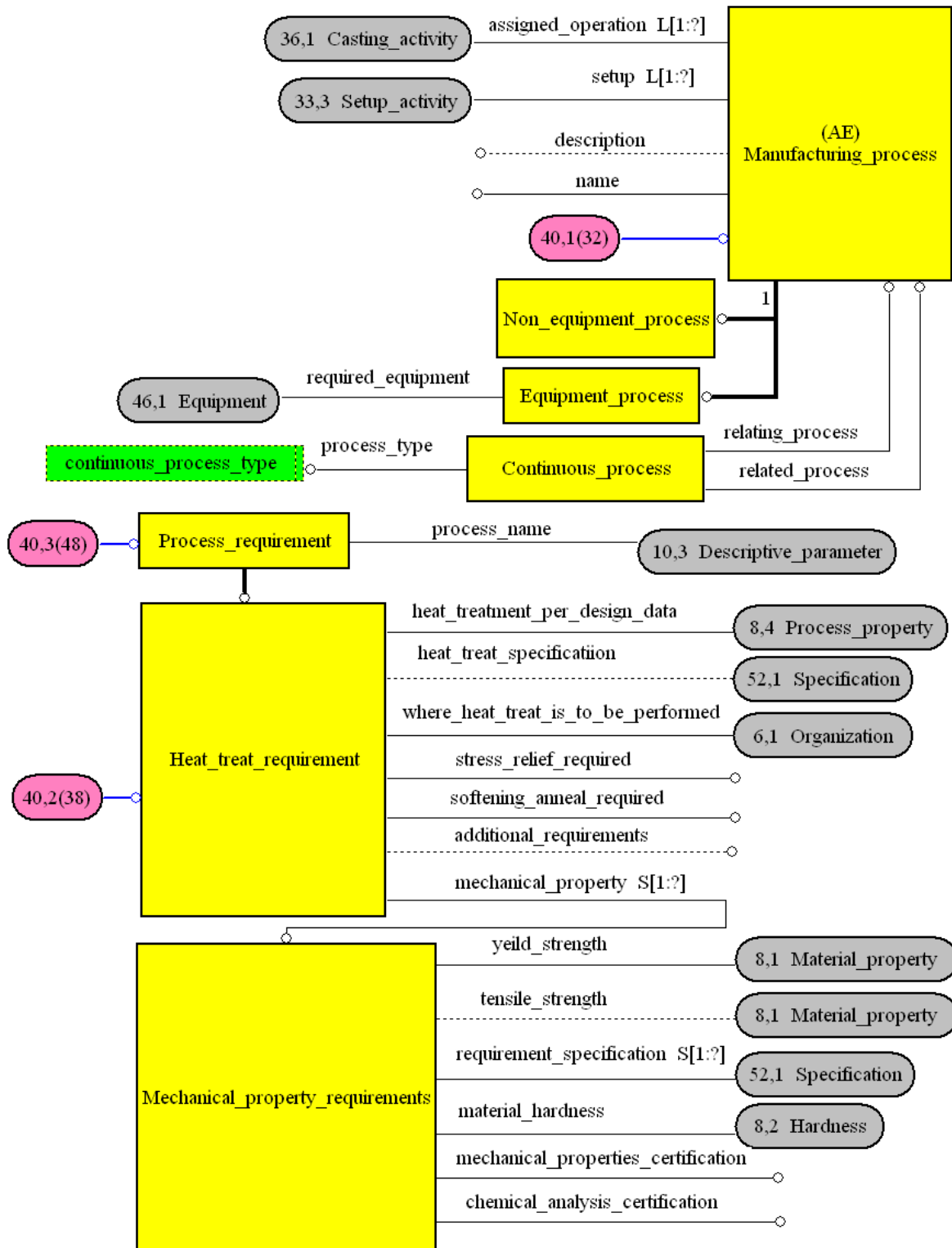


Figure G.40 — ARM diagram (40 of 53)



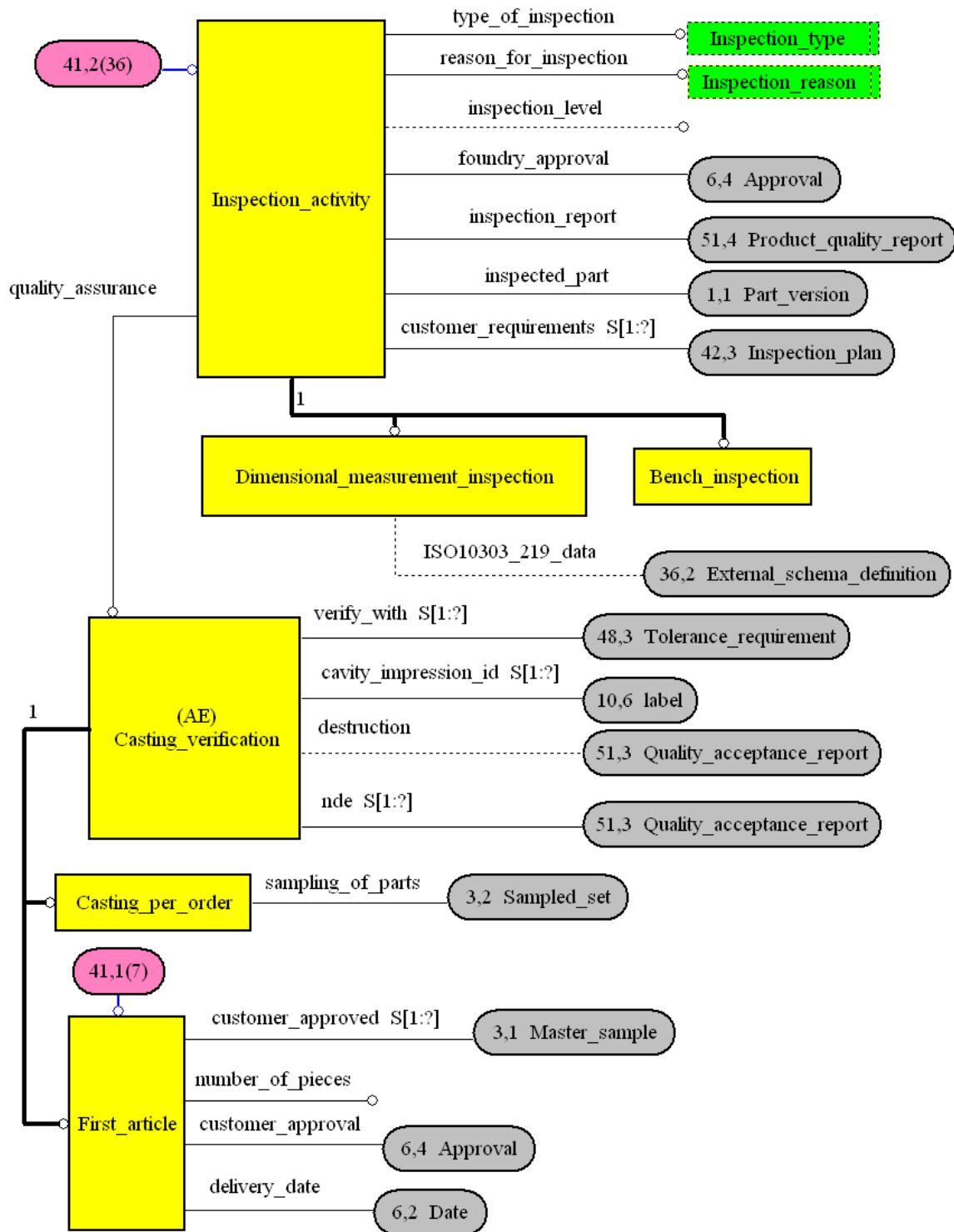
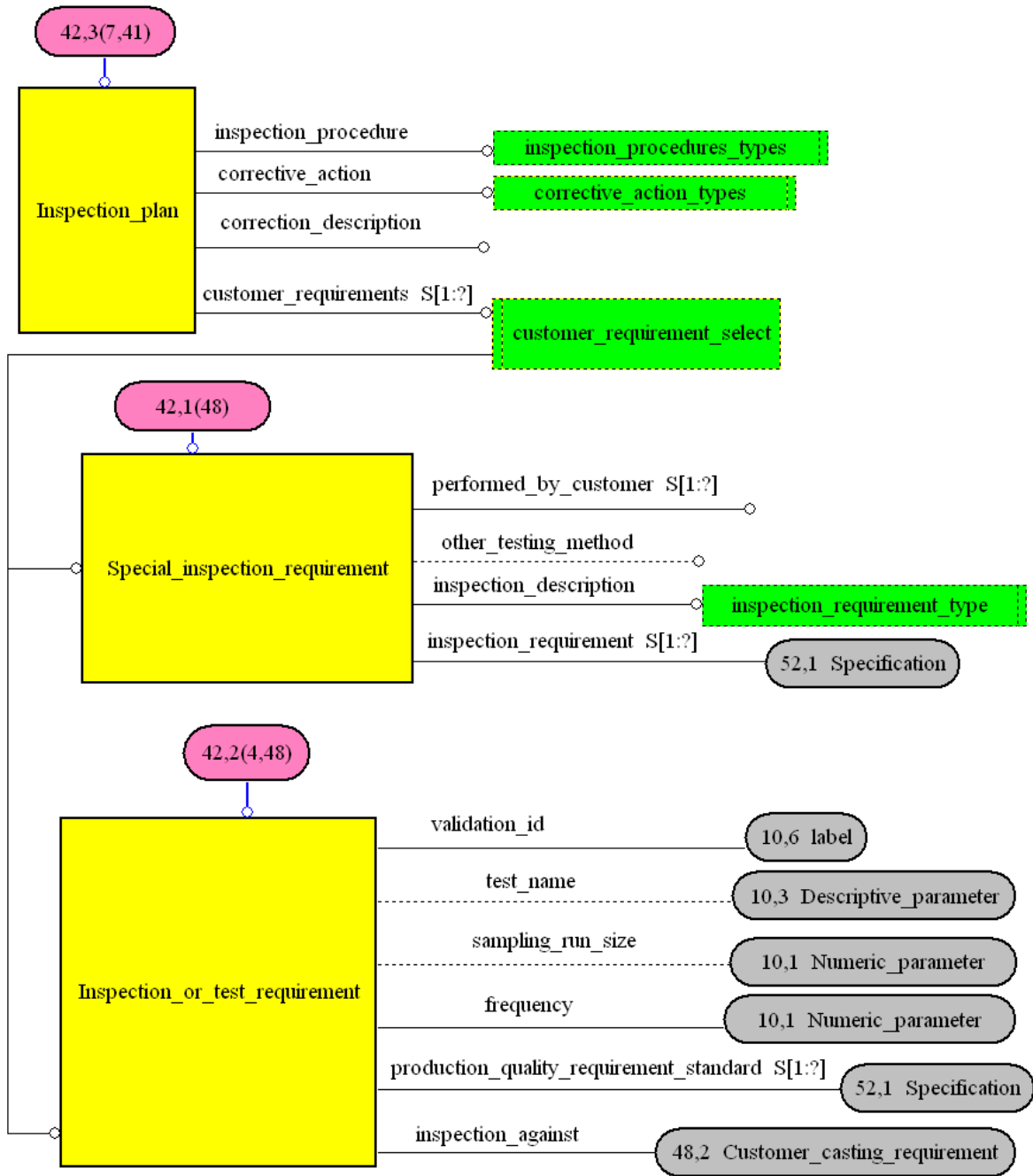


Figure G.41 — ARM diagram (41 of 53)



**Figure G.42 — ARM diagram (42 of 53)**

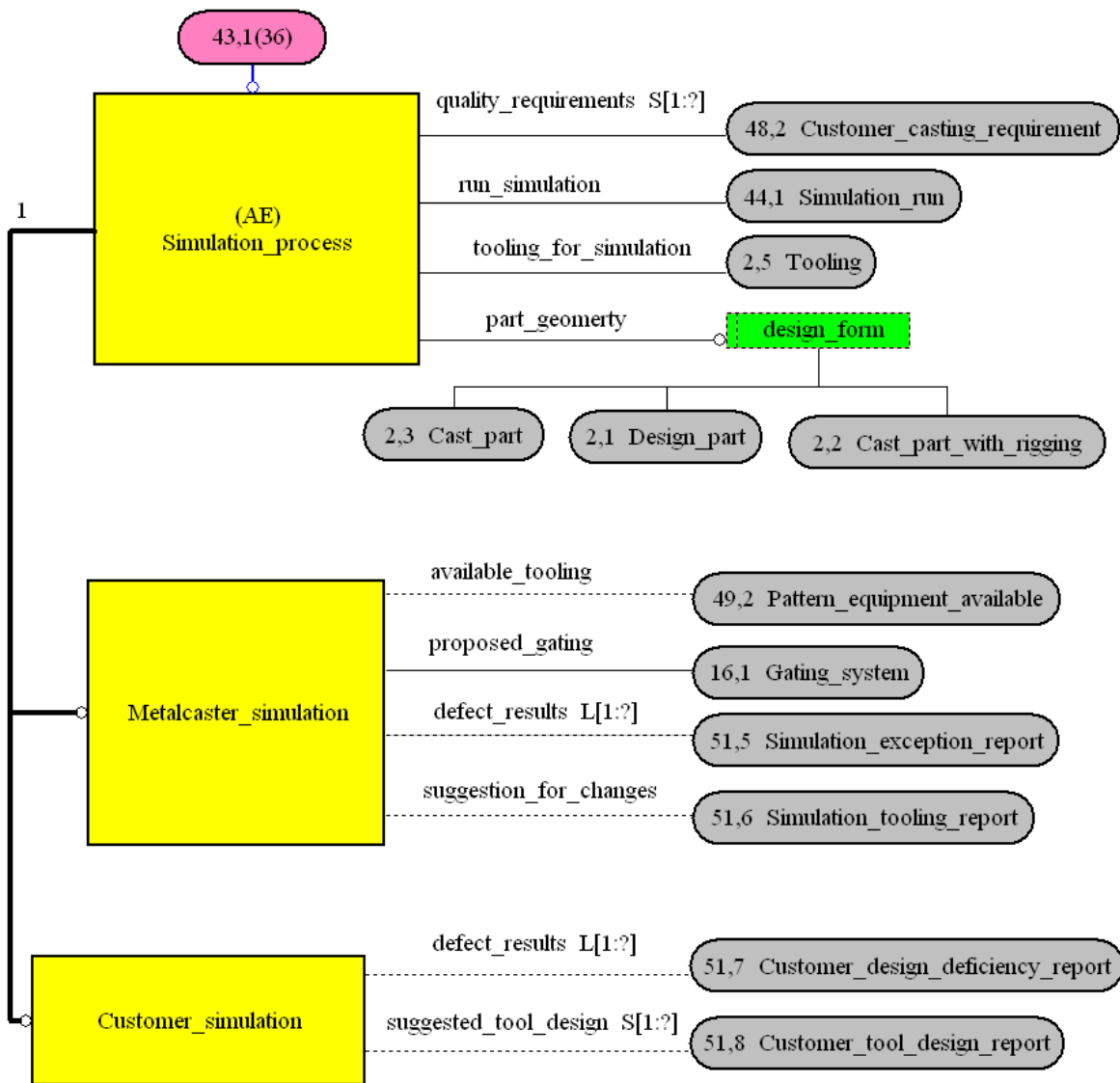


Figure G.43 — ARM diagram (43 of 53)

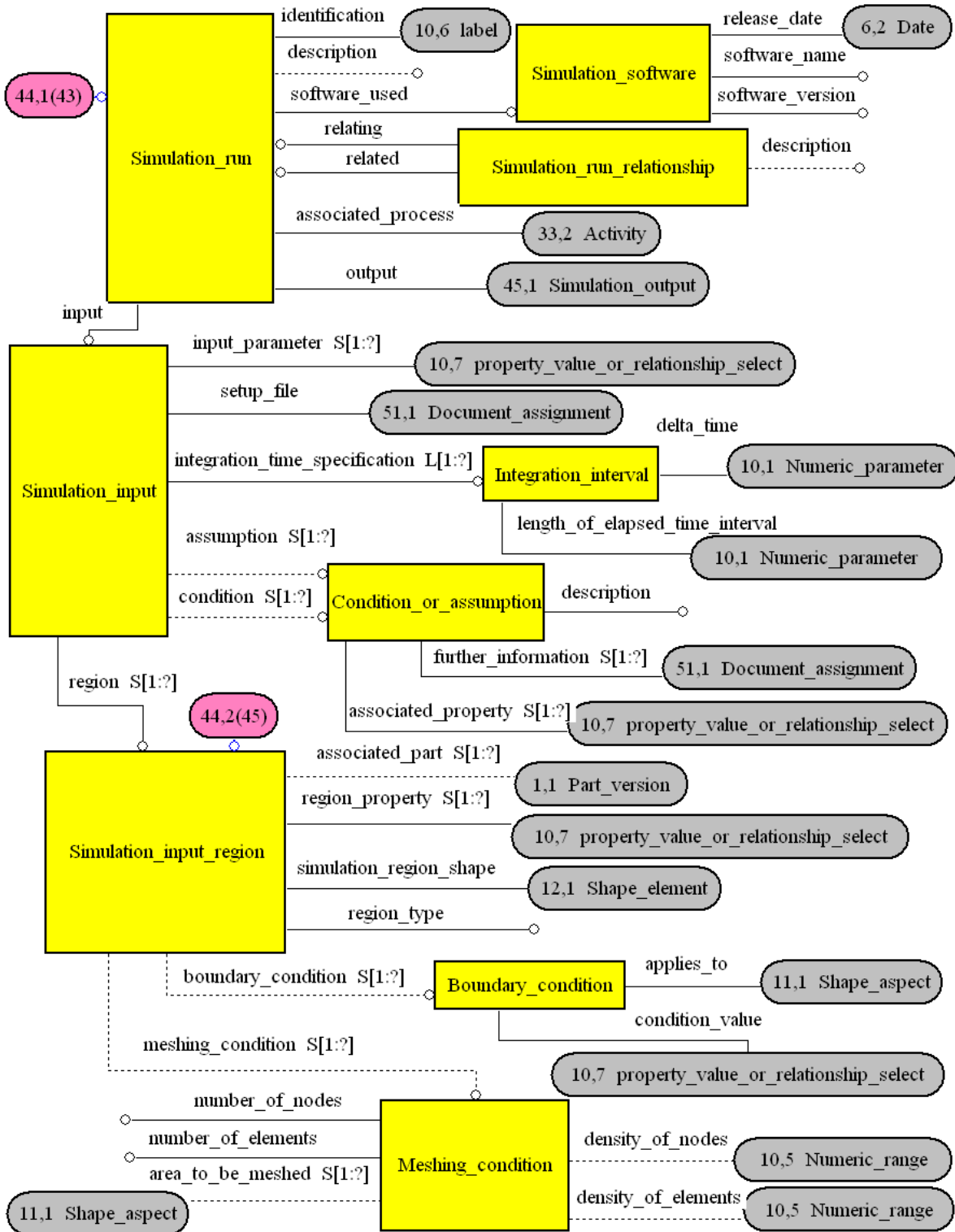


Figure G.44 — ARM diagram (44 of 53)

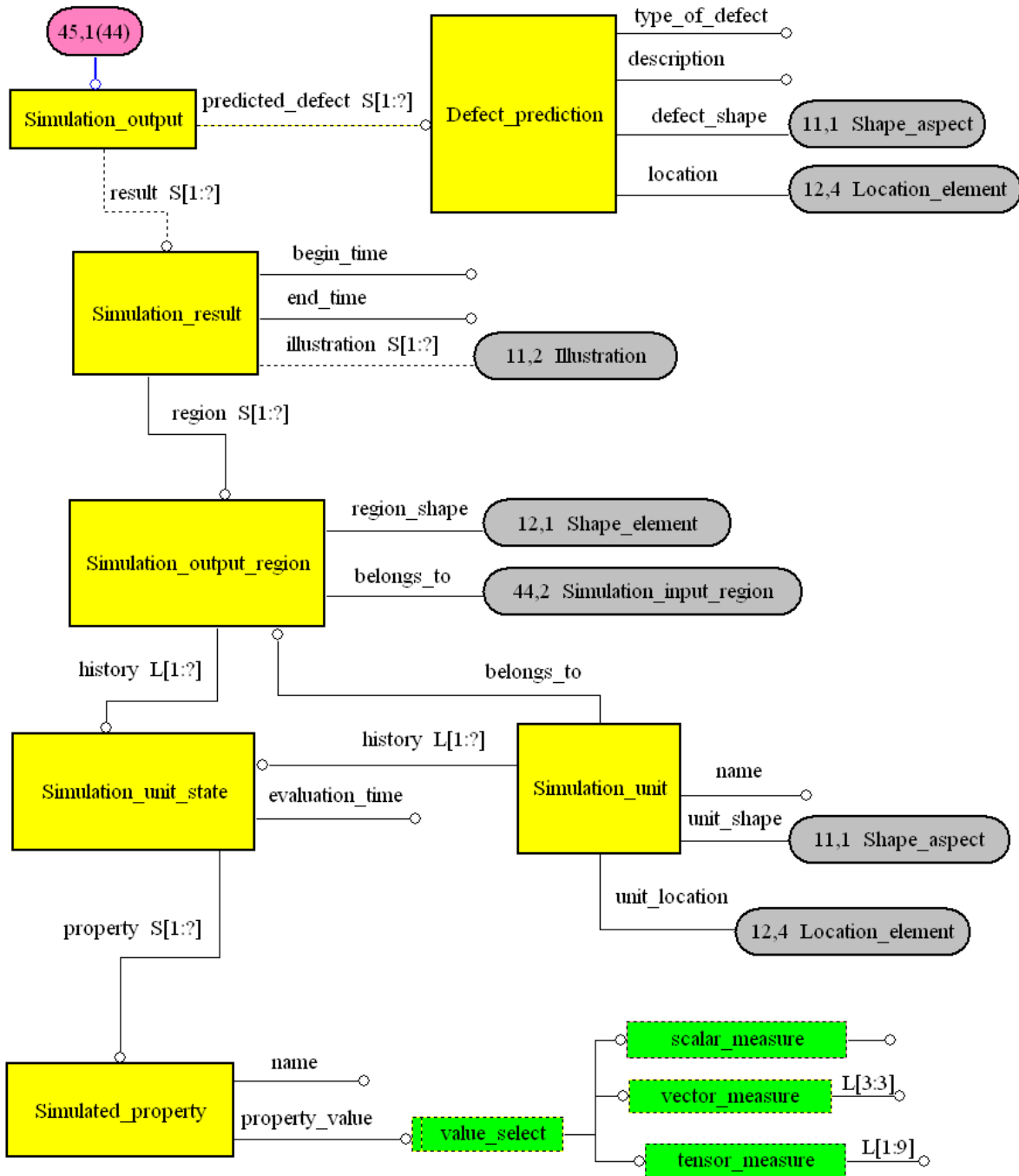


Figure G.45 — ARM diagram (45 of 53)

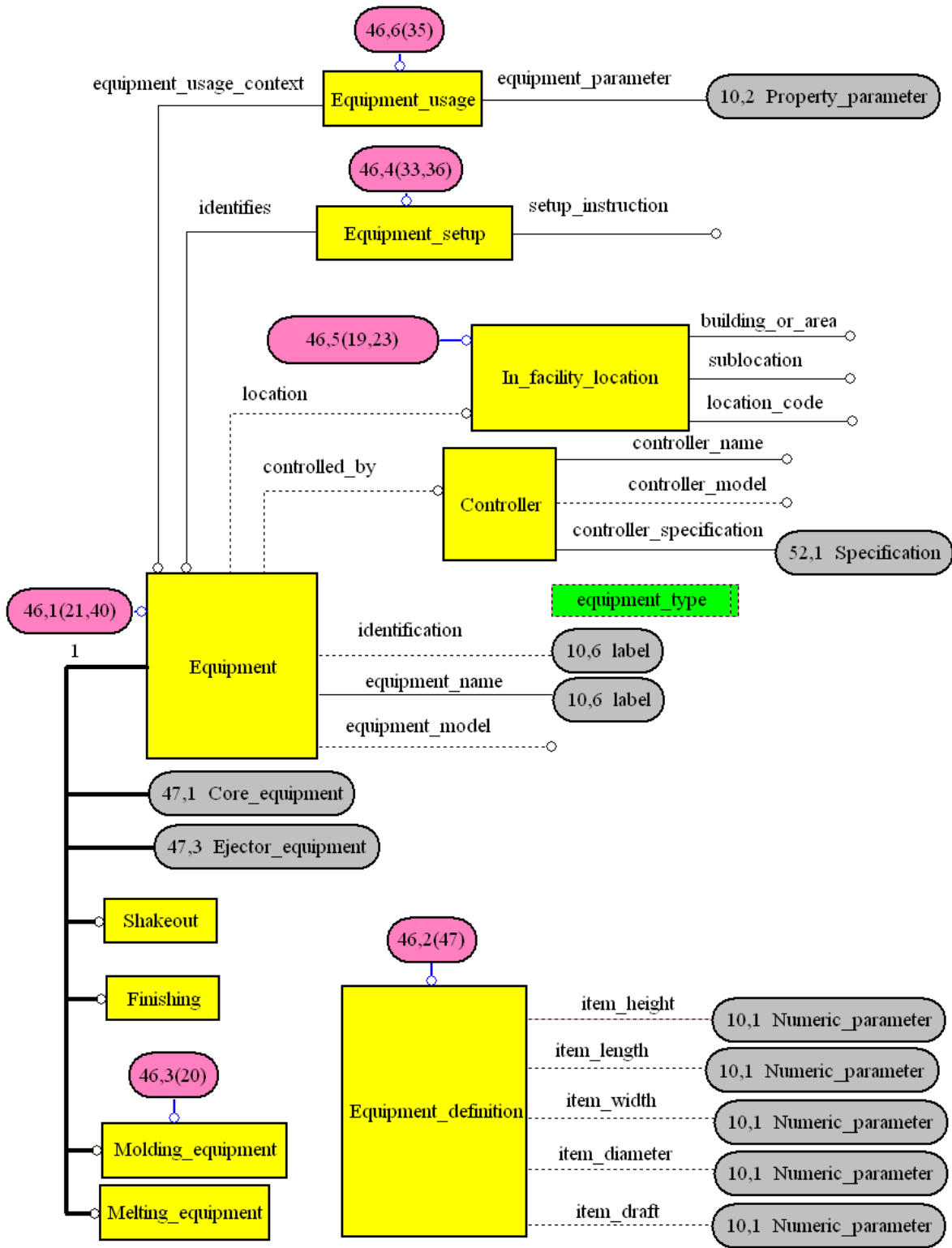


Figure G.46 — ARM diagram (46 of 53)

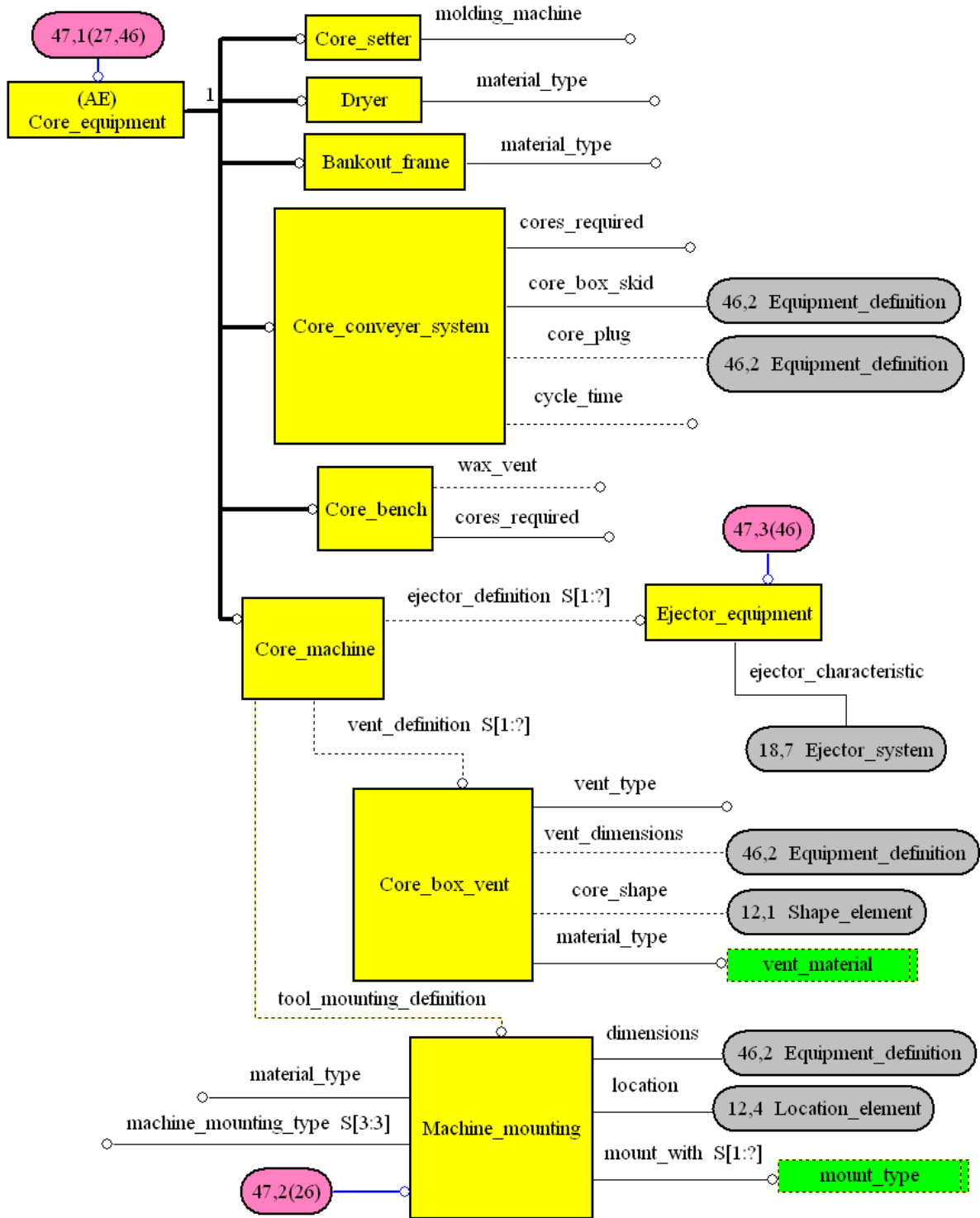


Figure G.47 — ARM diagram (47 of 53)

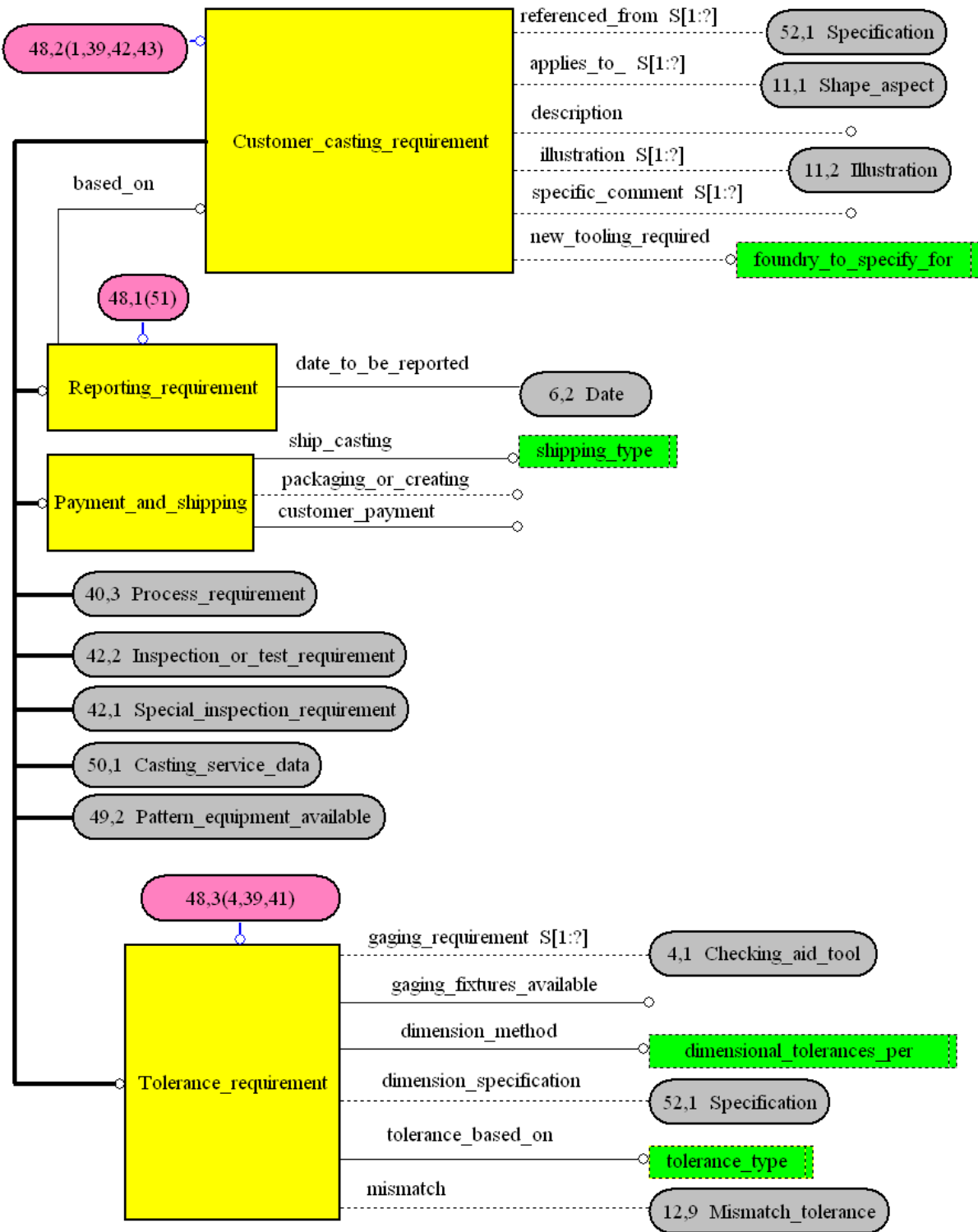
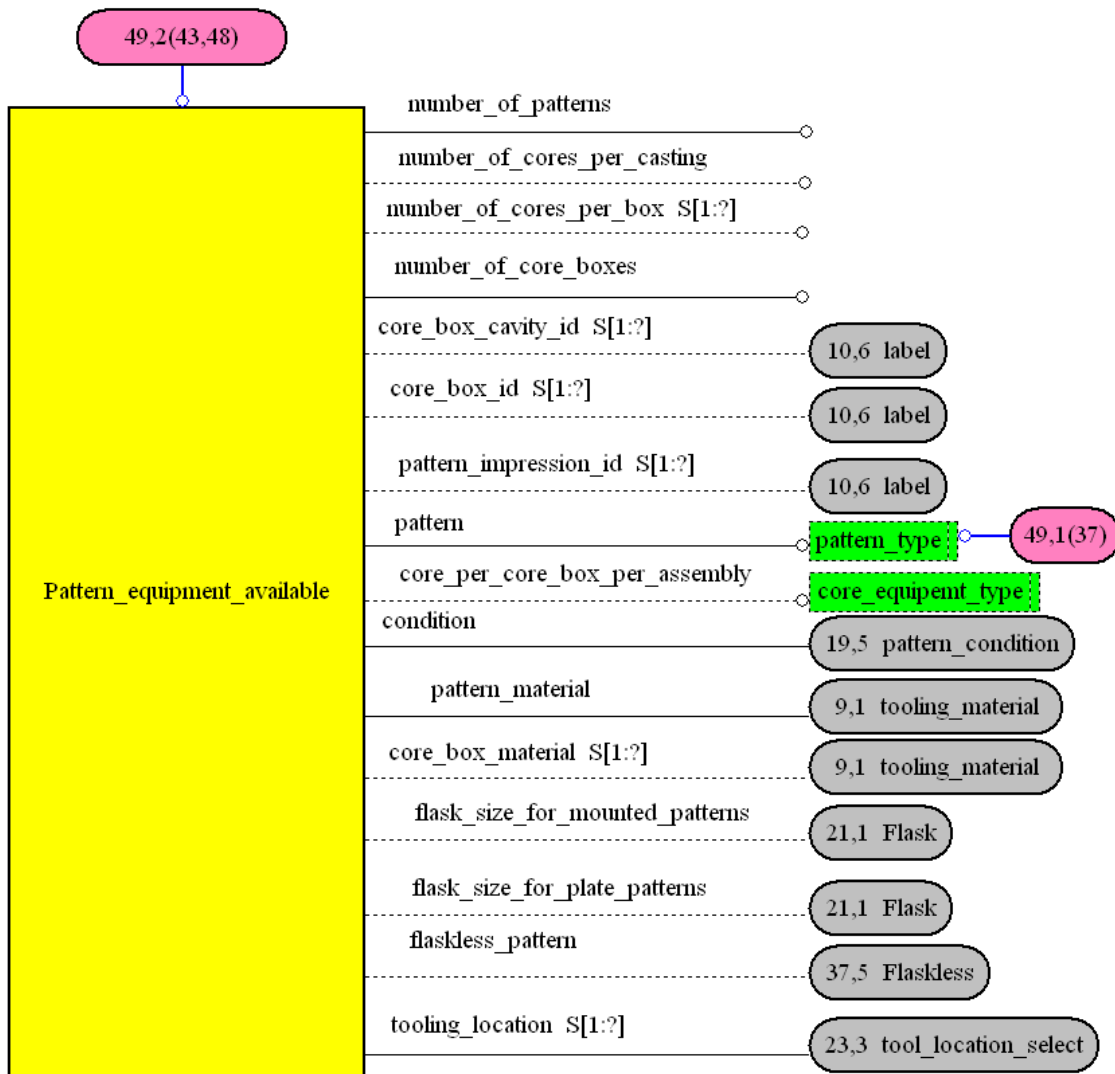


Figure G.48 — ARM diagram (48 of 53)





**Figure G.49 — ARM diagram (49 of 53)**

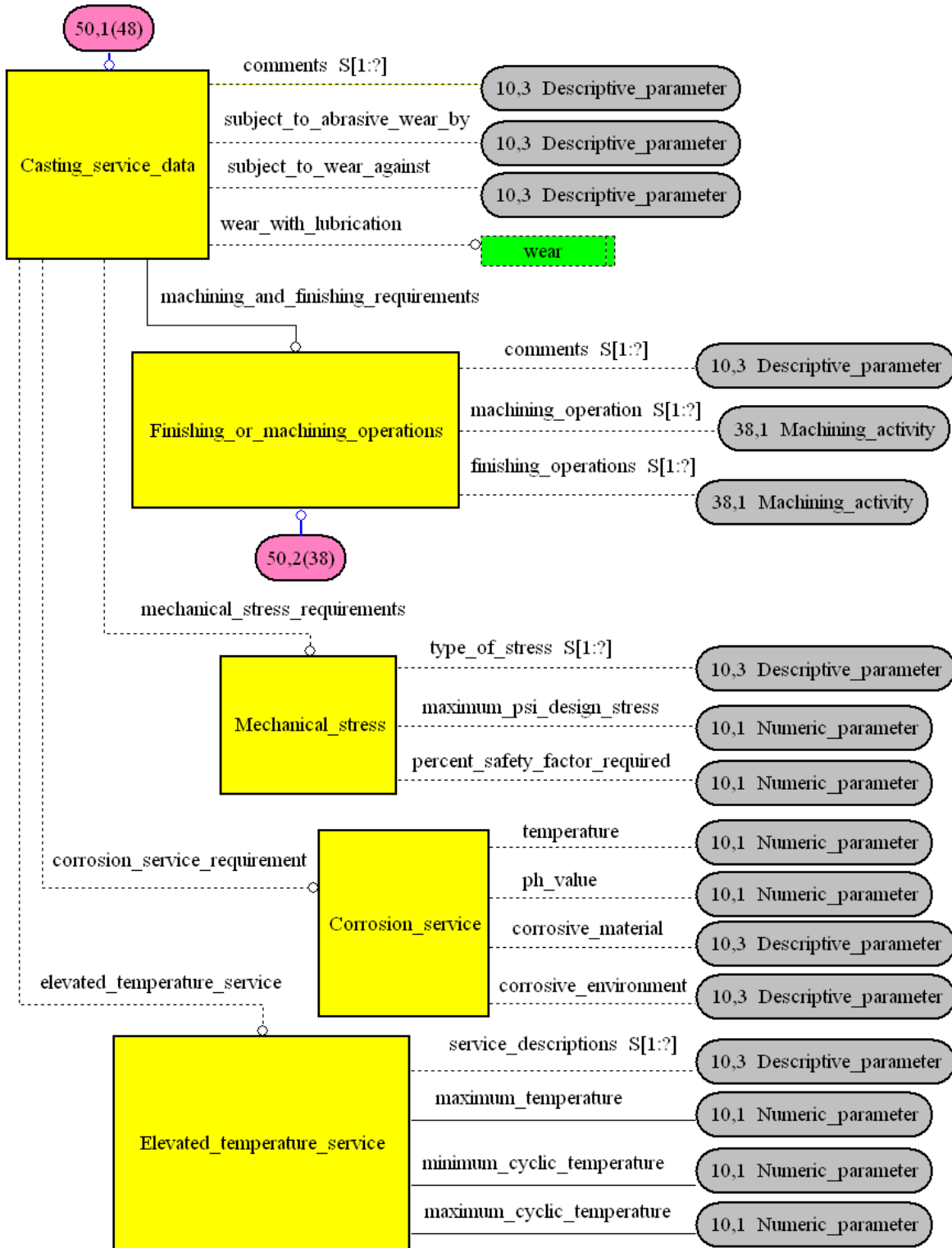


Figure G.50 — ARM diagram (50 of 53)

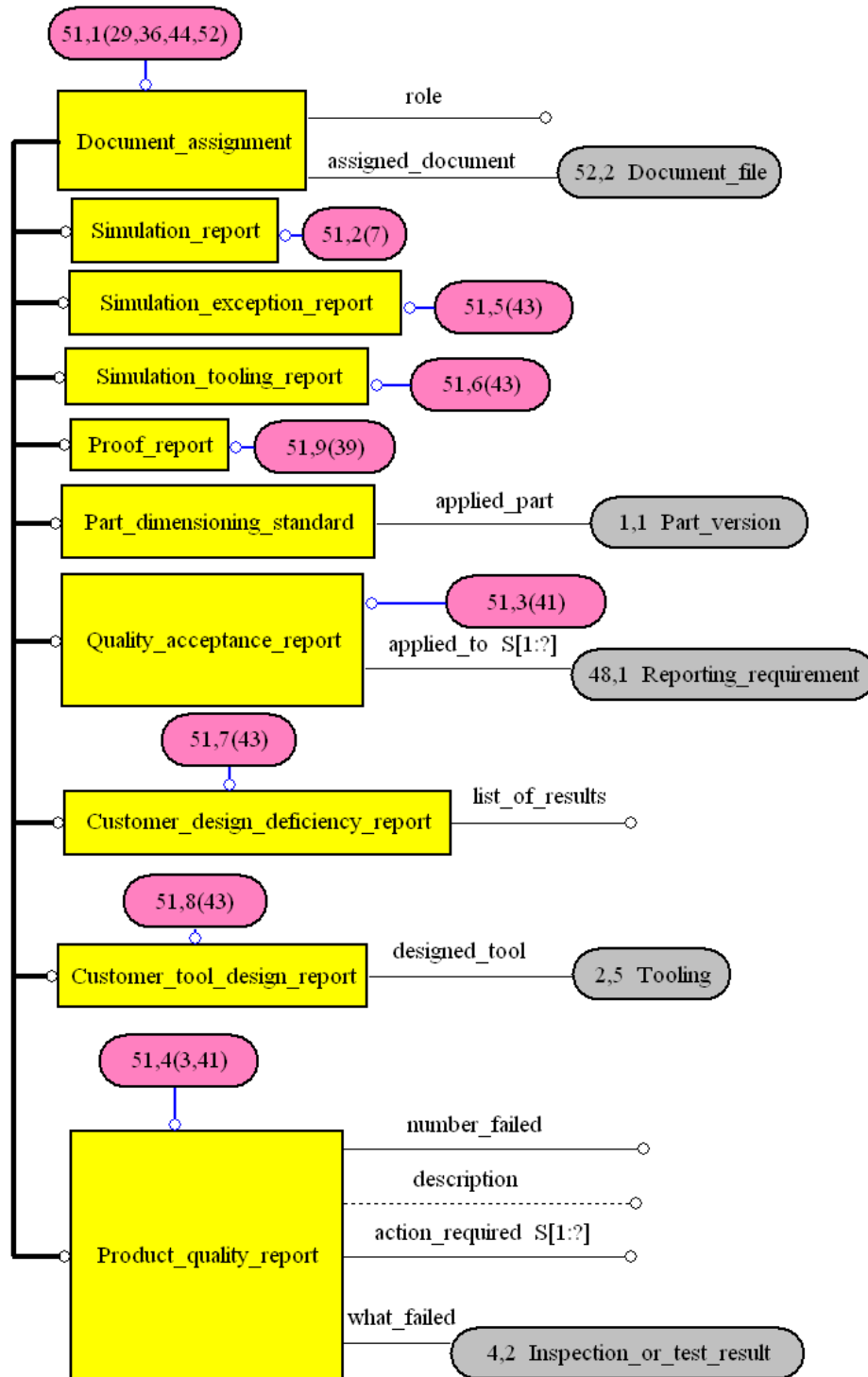


Figure G.51 — ARM diagram (51 of 53)

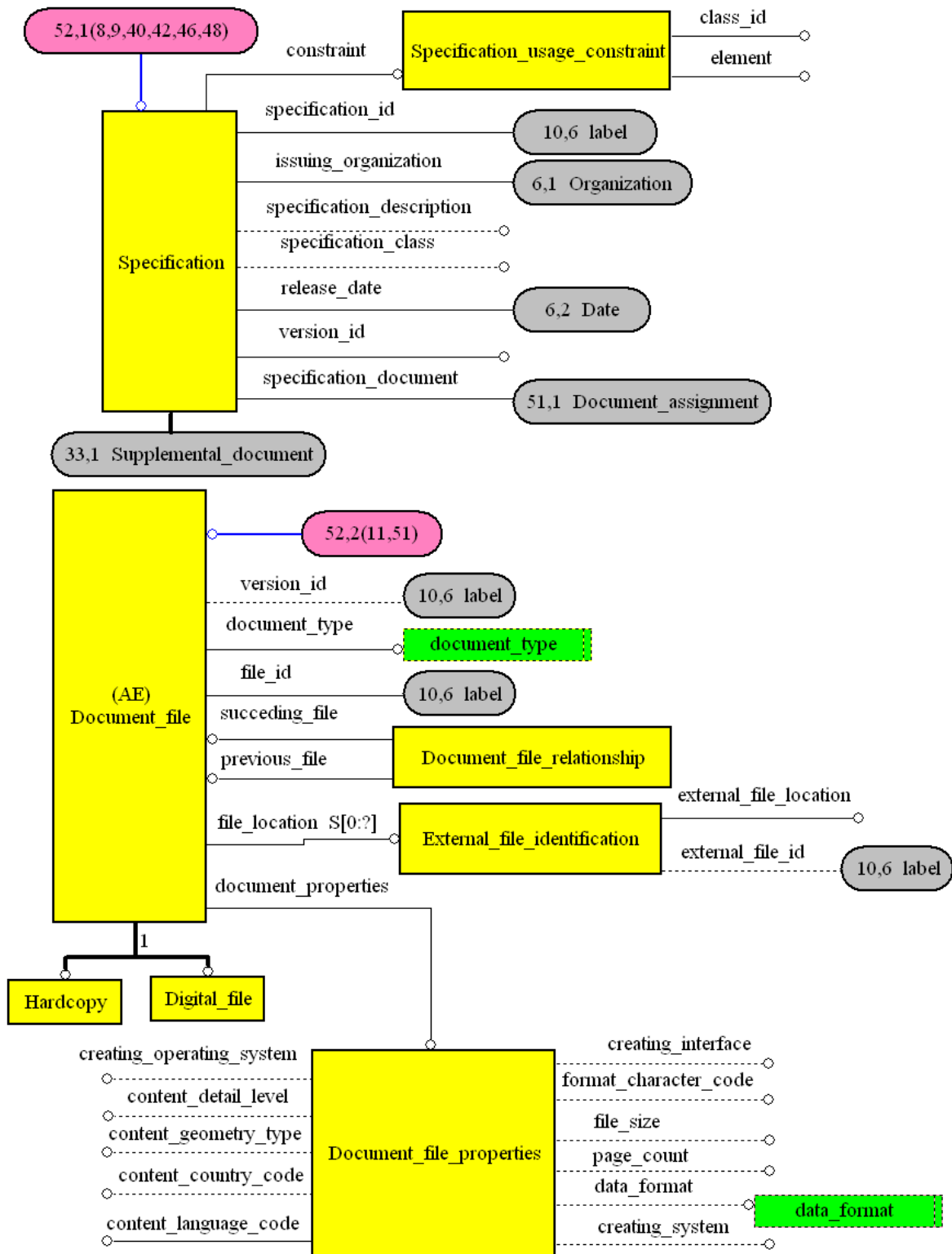
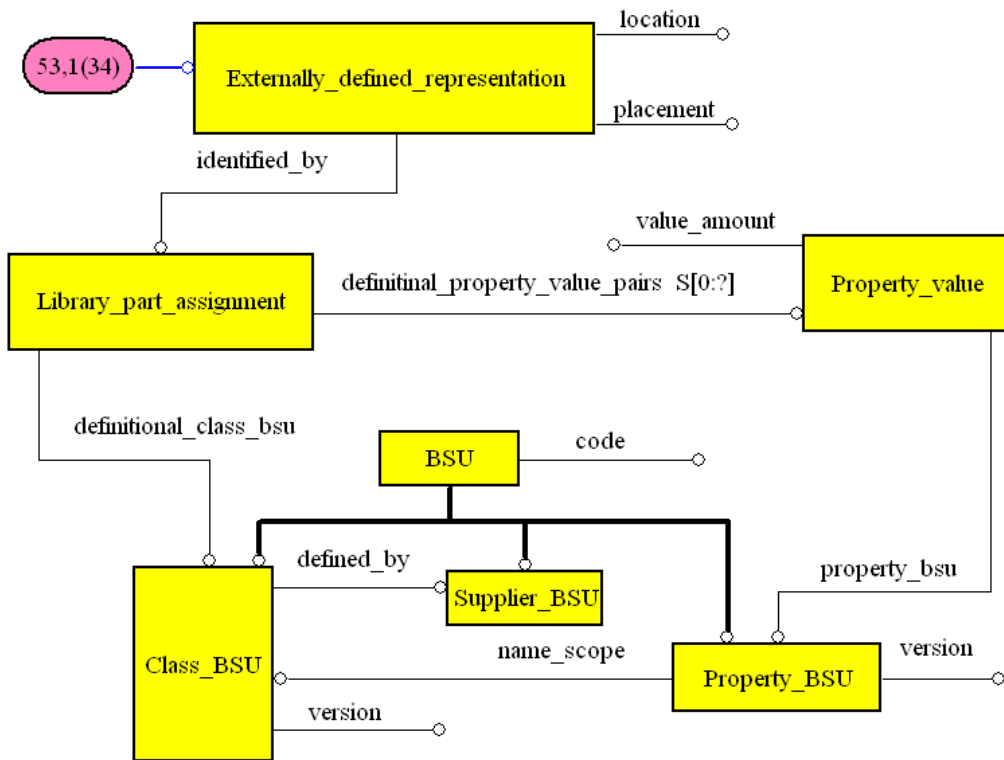


Figure G.52 — ARM diagram (52 of 53)



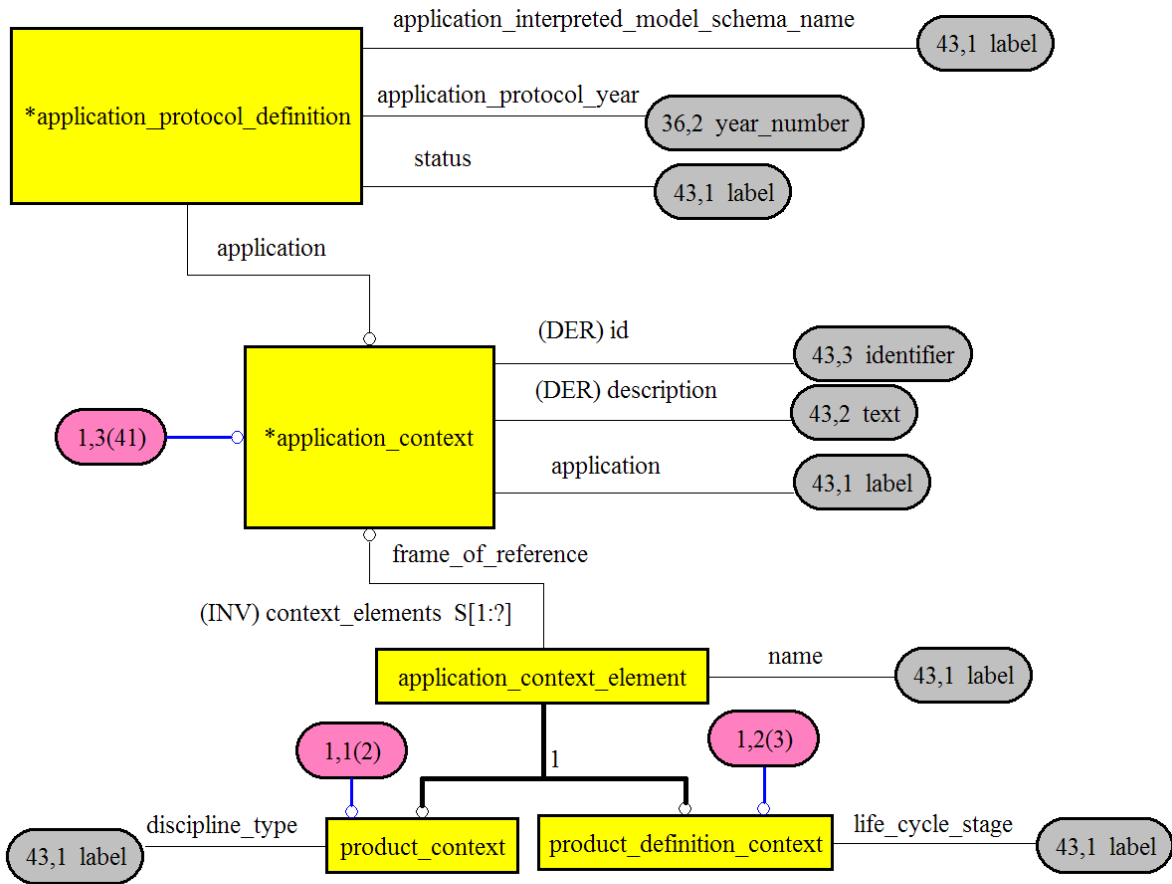
**Figure G.53 — ARM diagram (53 of 53)**

## **Annex H**

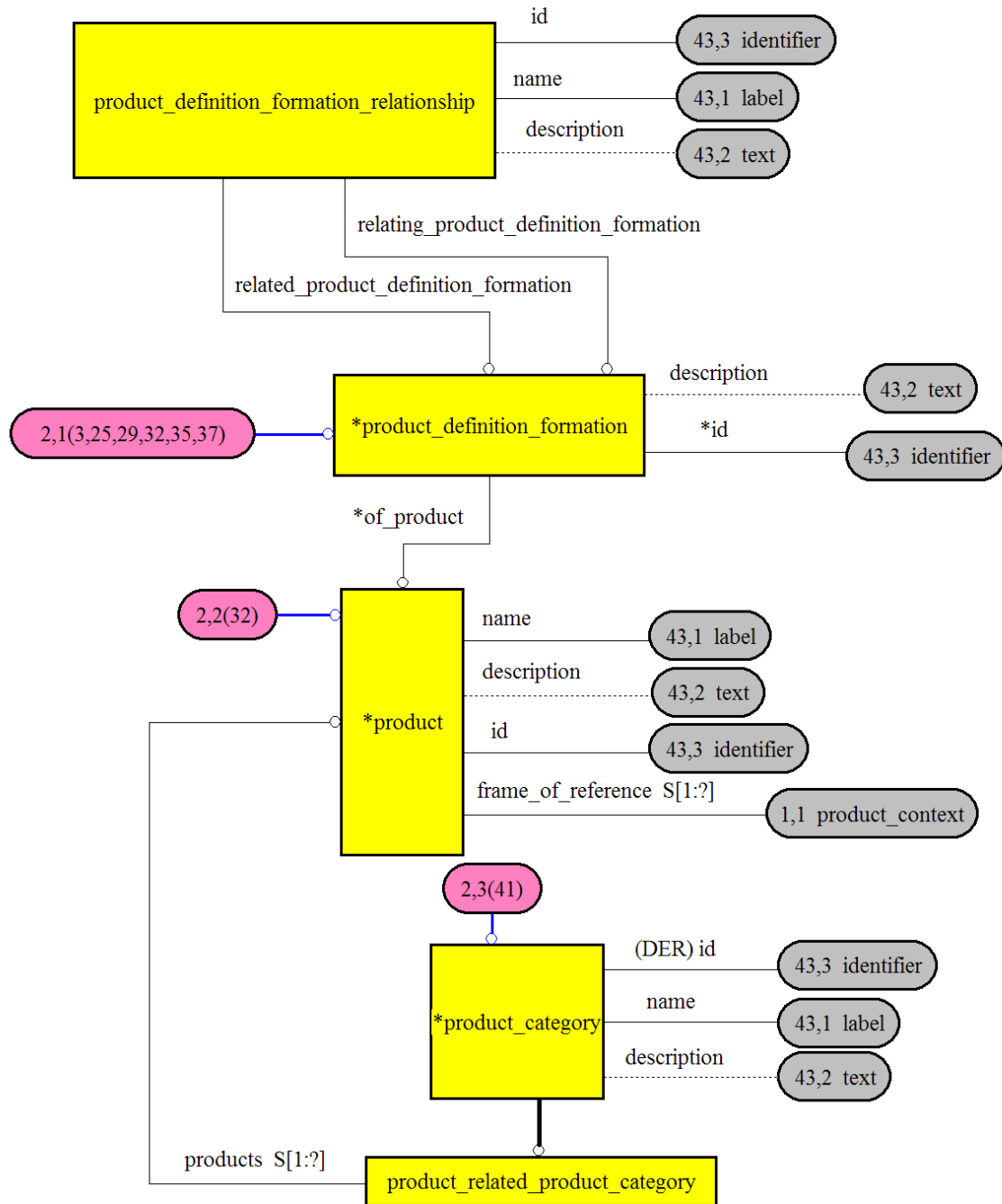
(informative)

### **AIM EXPRESS-G**

Figure H.1 through H.37 correspond to the AIM EXPRESS annotated listing given in annex A. The figures use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in annex A of ISO 10303-11:2004.



**Figure H.1 — AIM EXPRESS-G diagram application\_context**



**Figure H.2 — AIM EXPRESS-G diagram product**



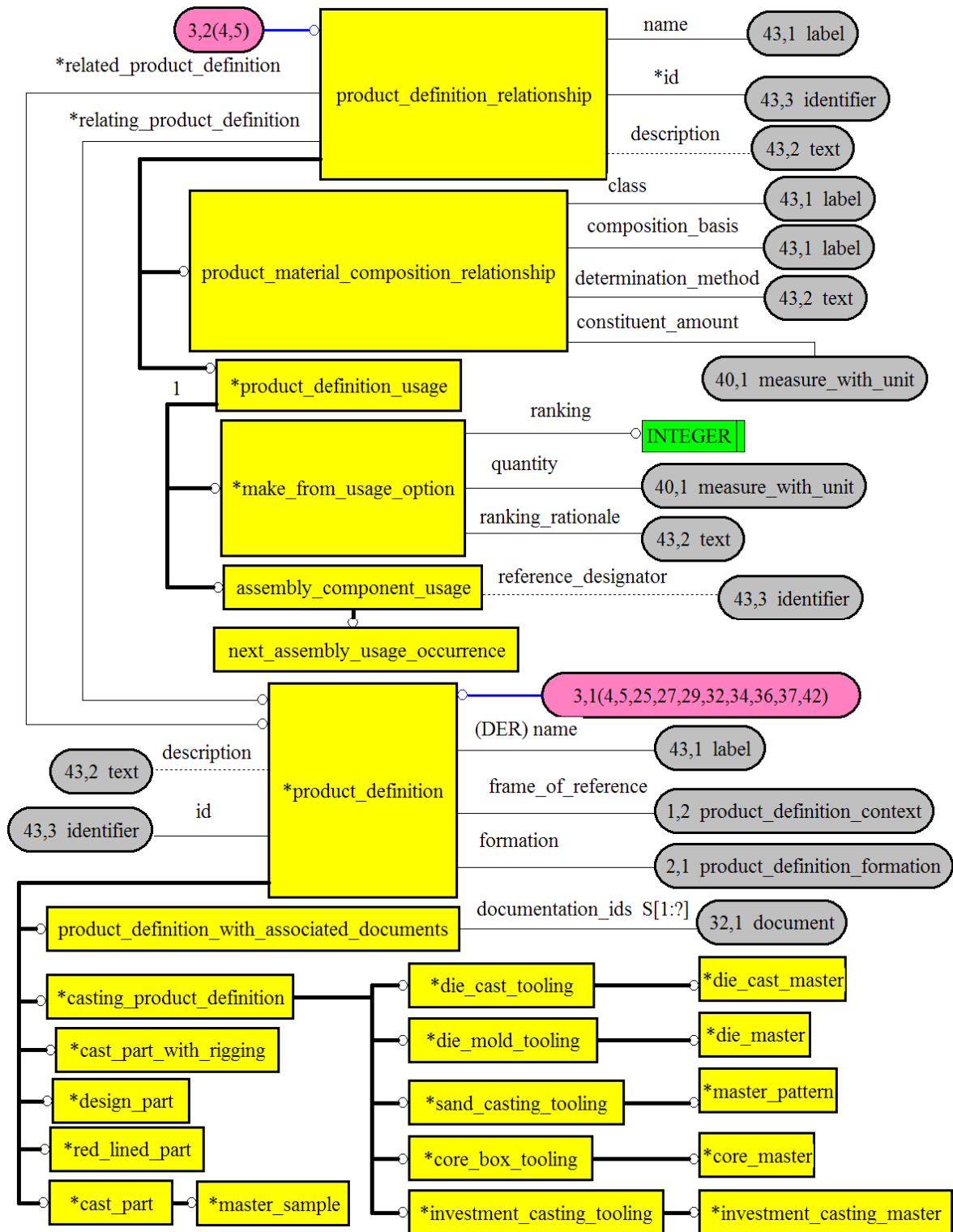


Figure H.3 — AIM EXPRESS-G diagram product\_definition

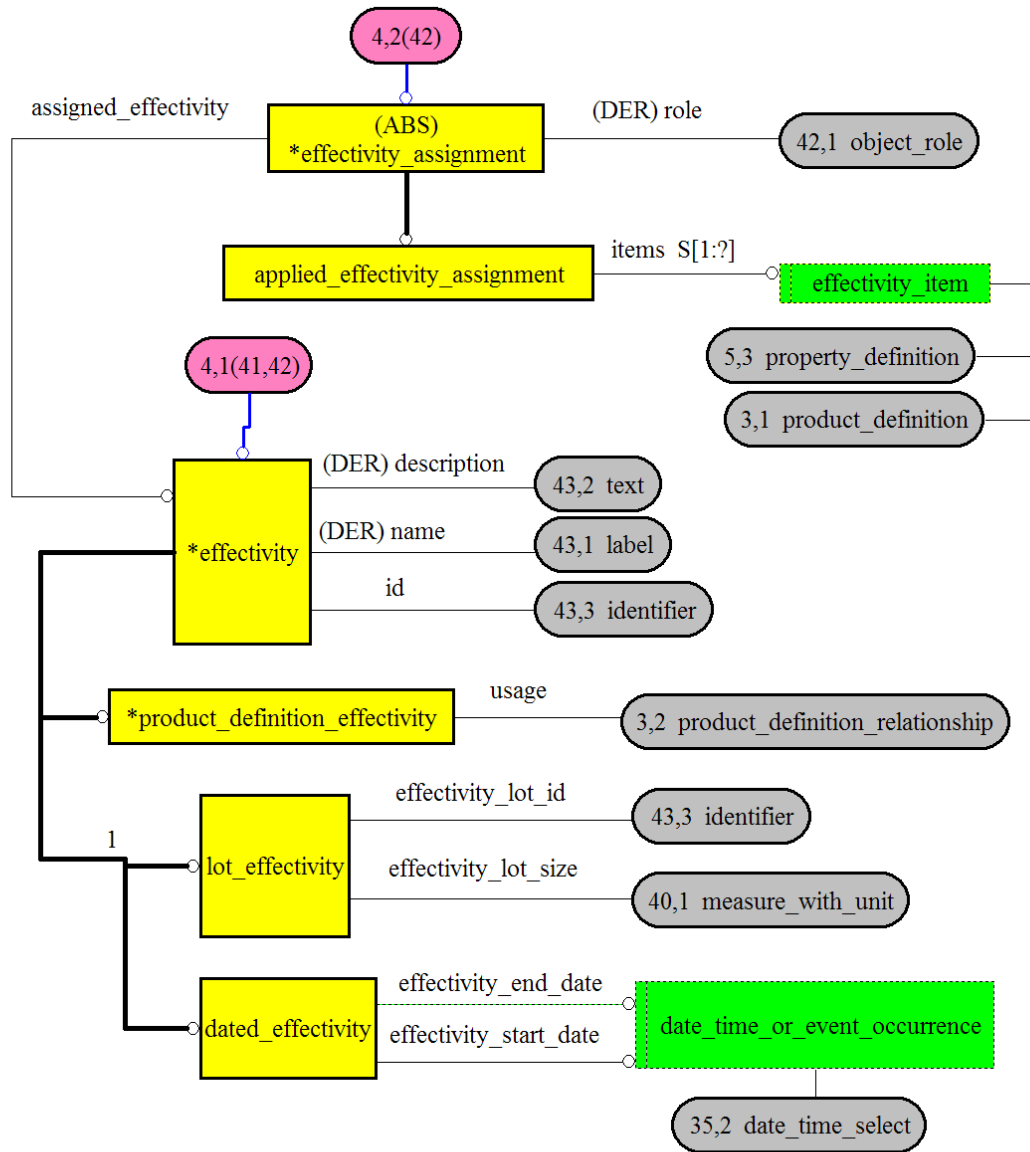
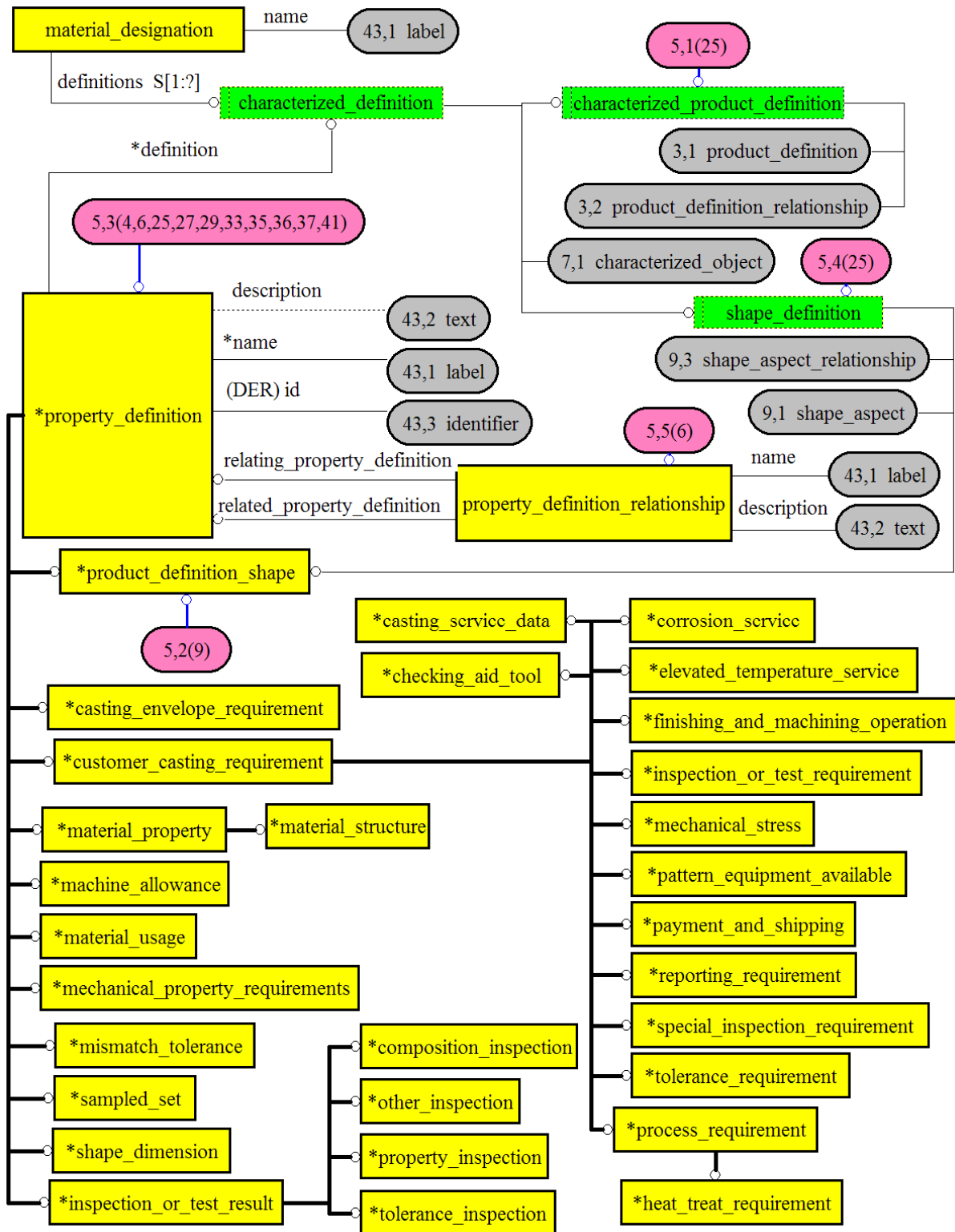
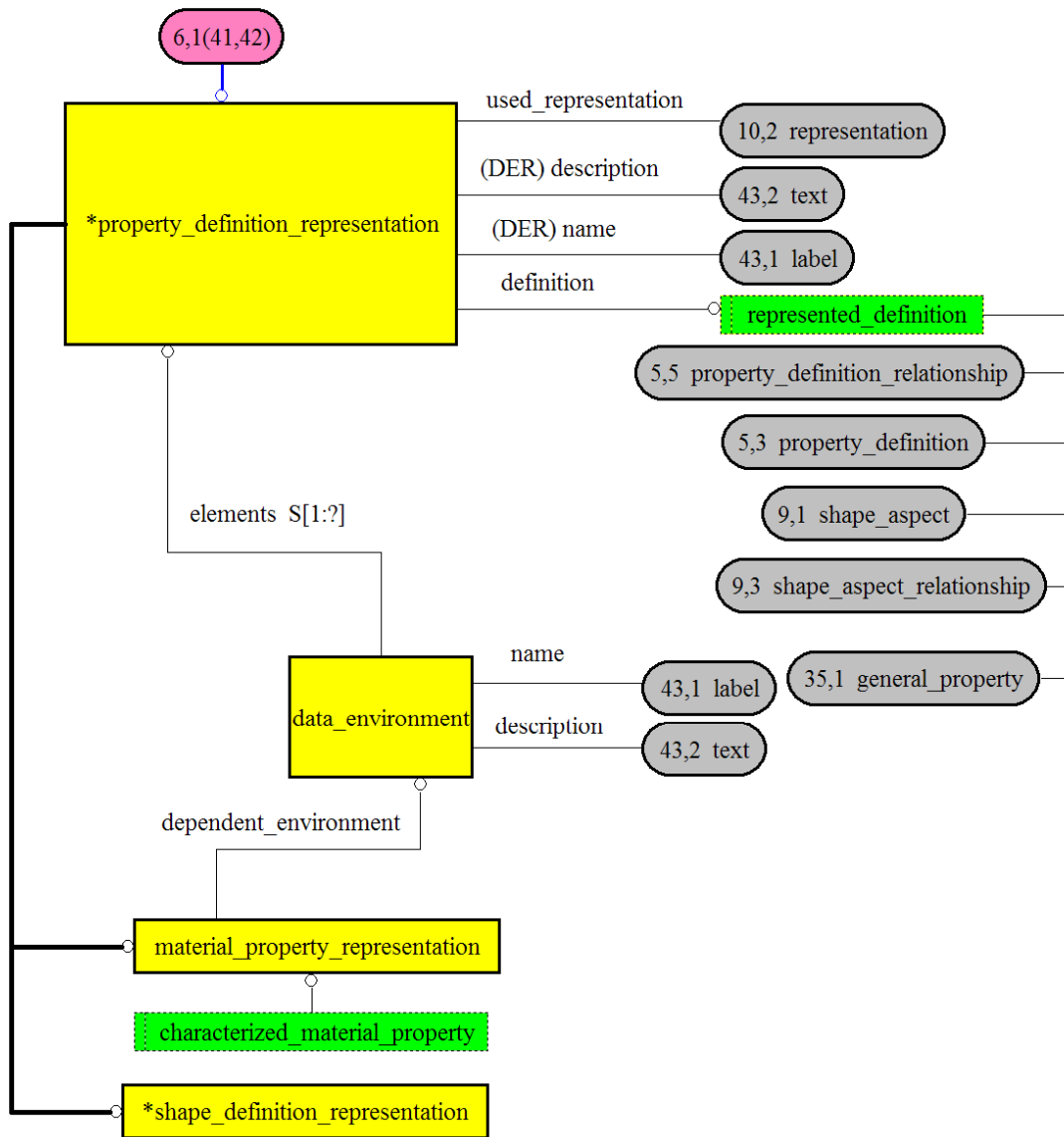


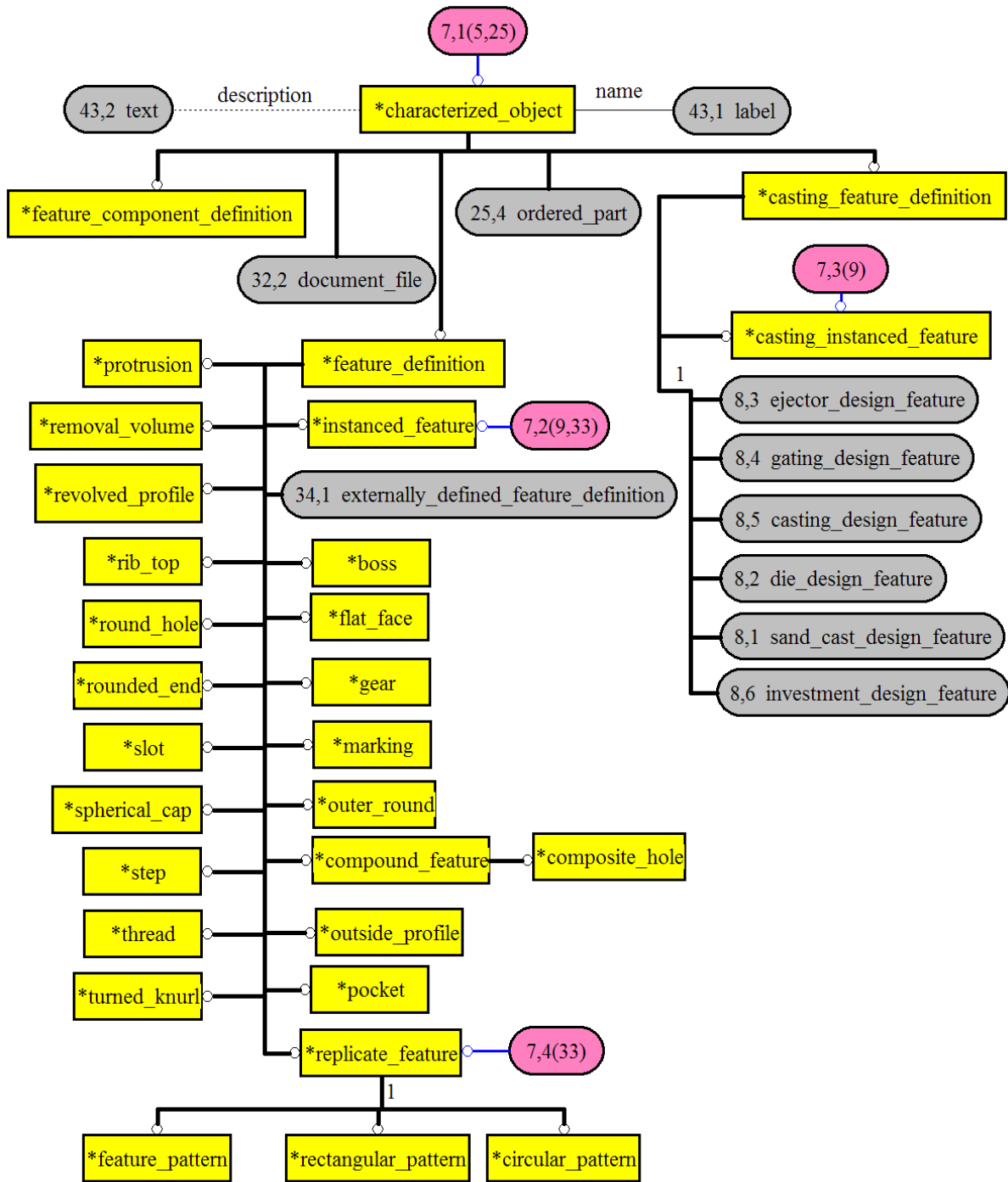
Figure H.4 — AIM EXPRESS-G diagram effectivity



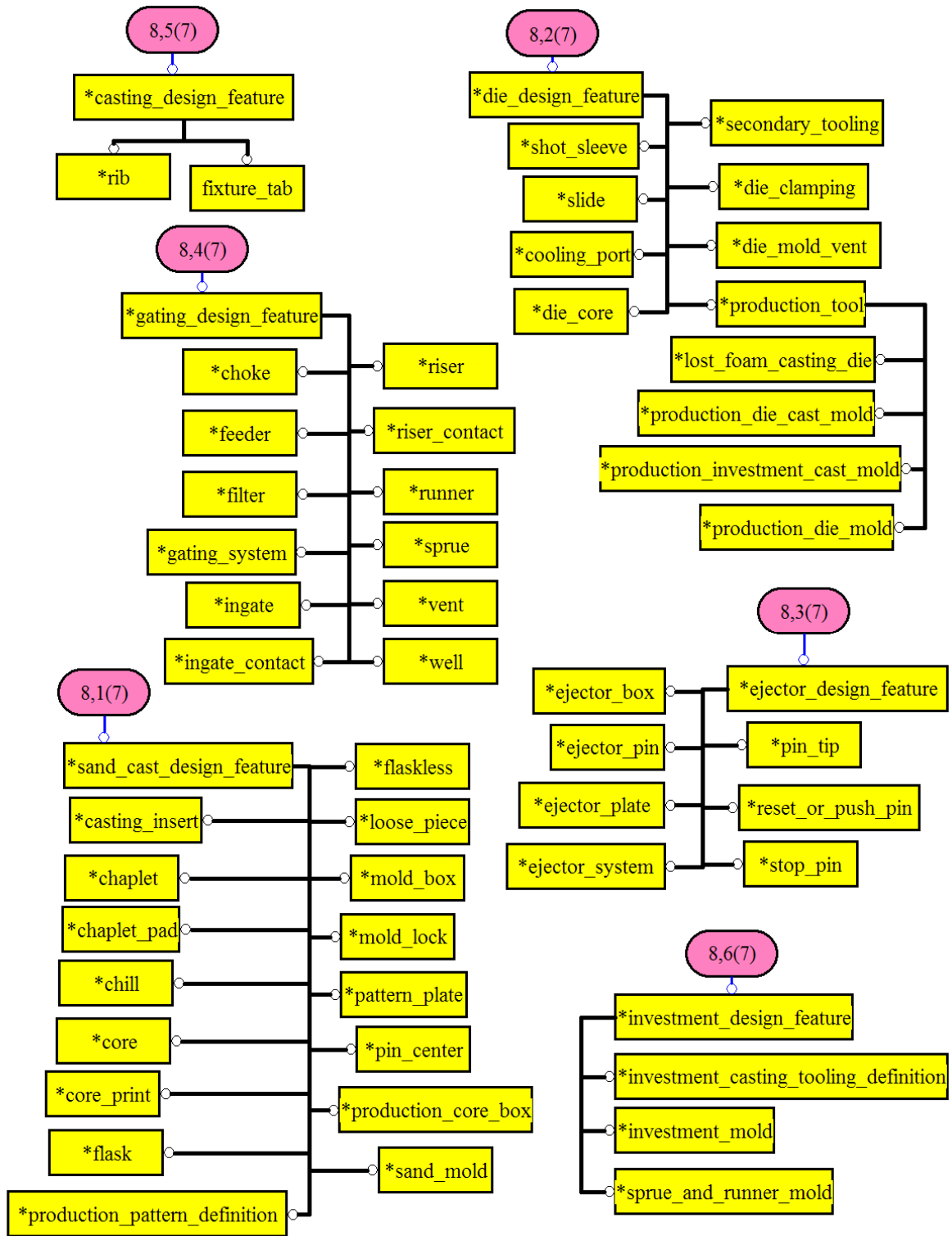
**Figure H.5 — AIM EXPRESS-G diagram property\_definition**



**Figure H.6 — AIM EXPRESS-G diagram  
property\_definition\_representation**



**Figure H.7 — AIM EXPRESS-G diagram characterized\_object**



**Figure H.8 — AIM EXPRESS-G diagram casting\_feature\_definition**

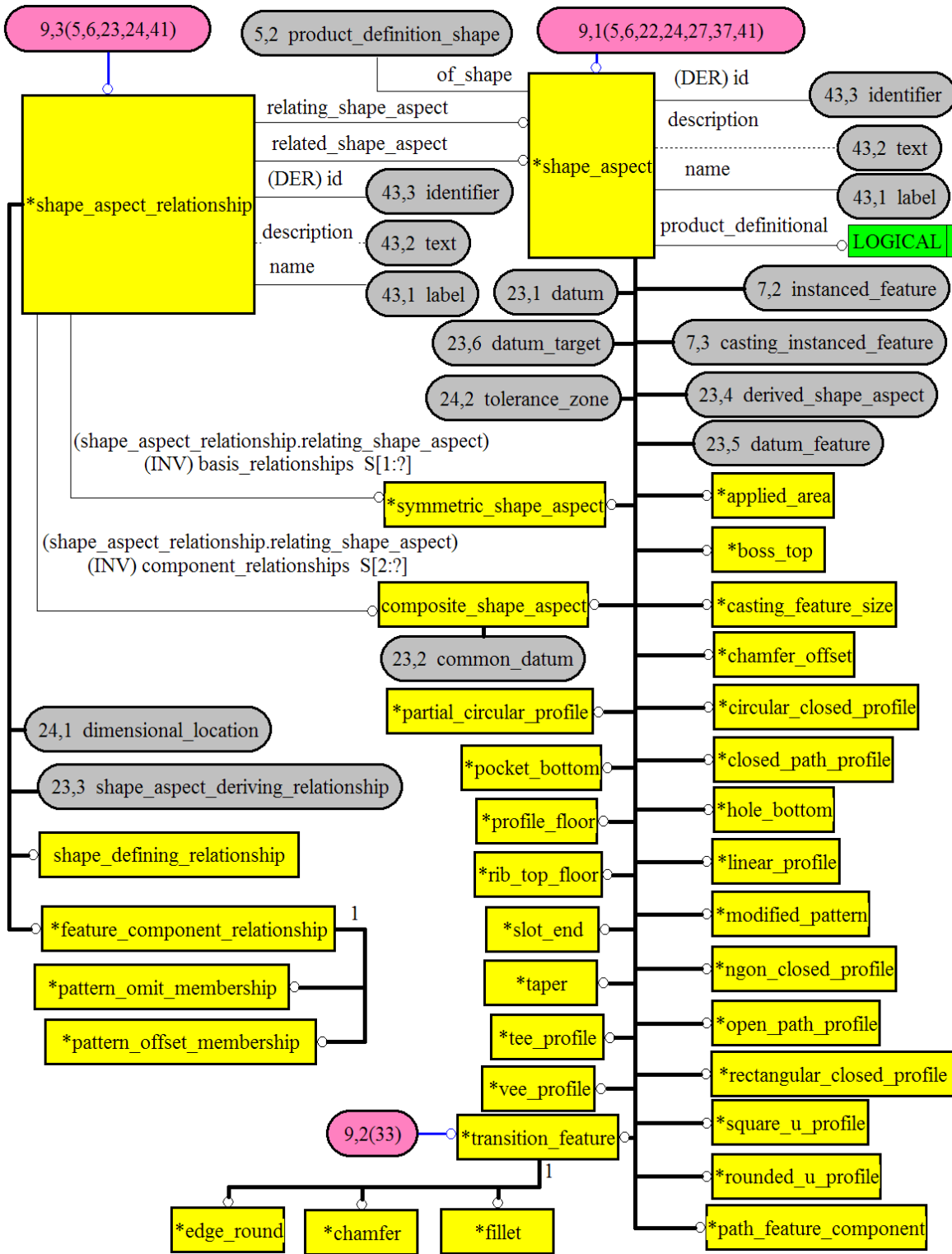
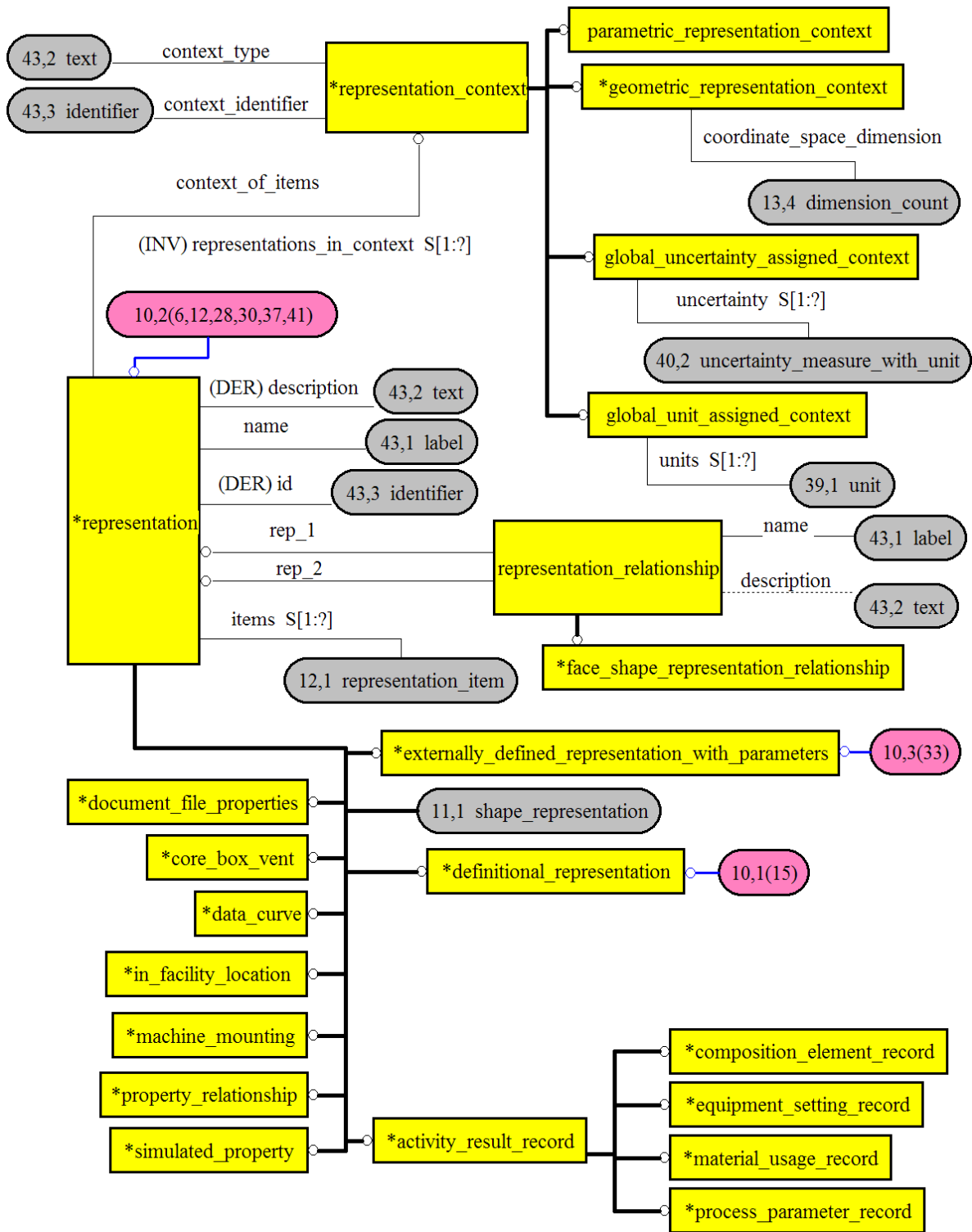
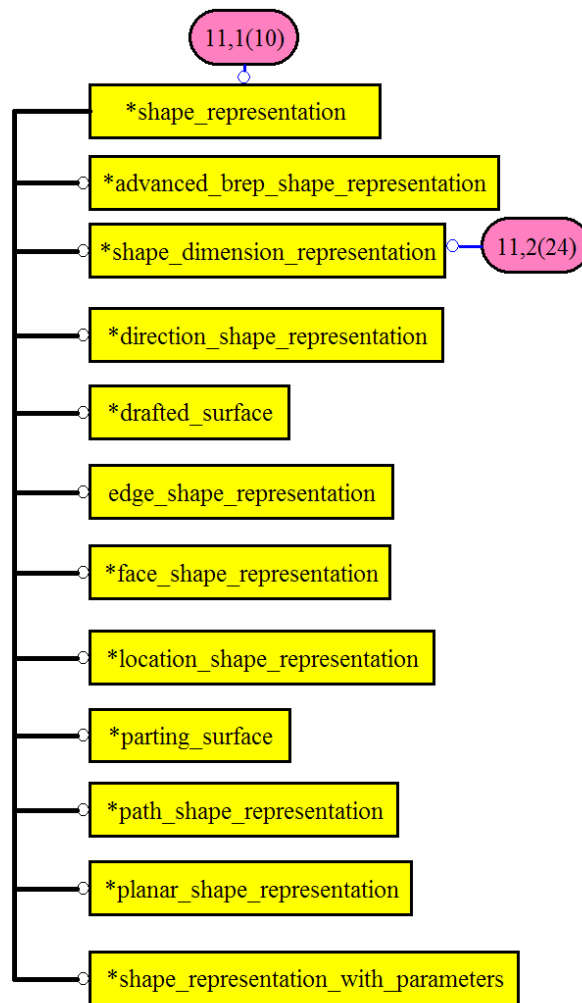


Figure H.9 — AIM EXPRESS-G diagram shape\_aspect

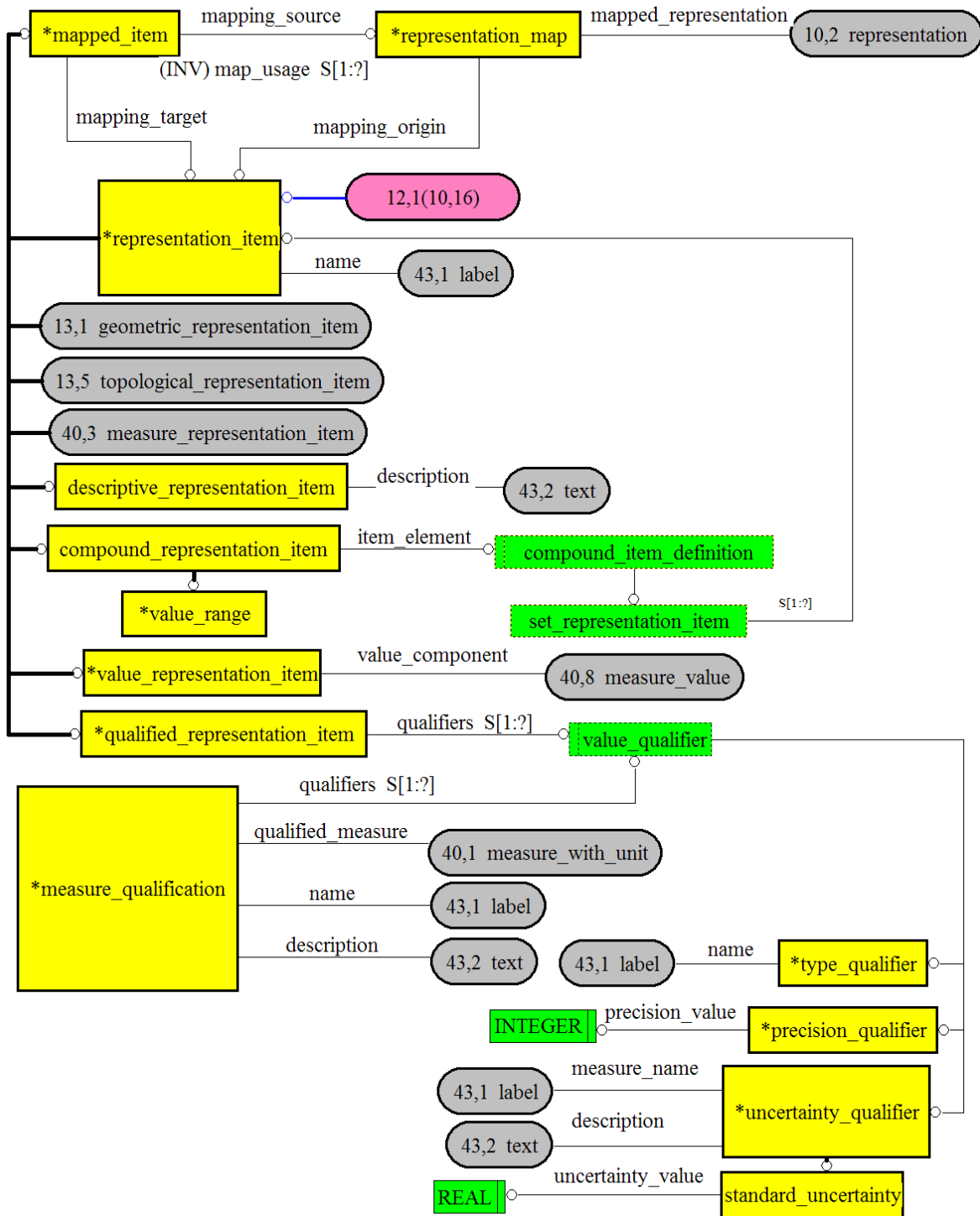


**Figure H.10 — AIM EXPRESS-G diagram representation**





**Figure H.11 — AIM EXPRESS-G diagram shape\_representation**



**Figure H.12 — AIM EXPRESS-G diagram representation\_item**

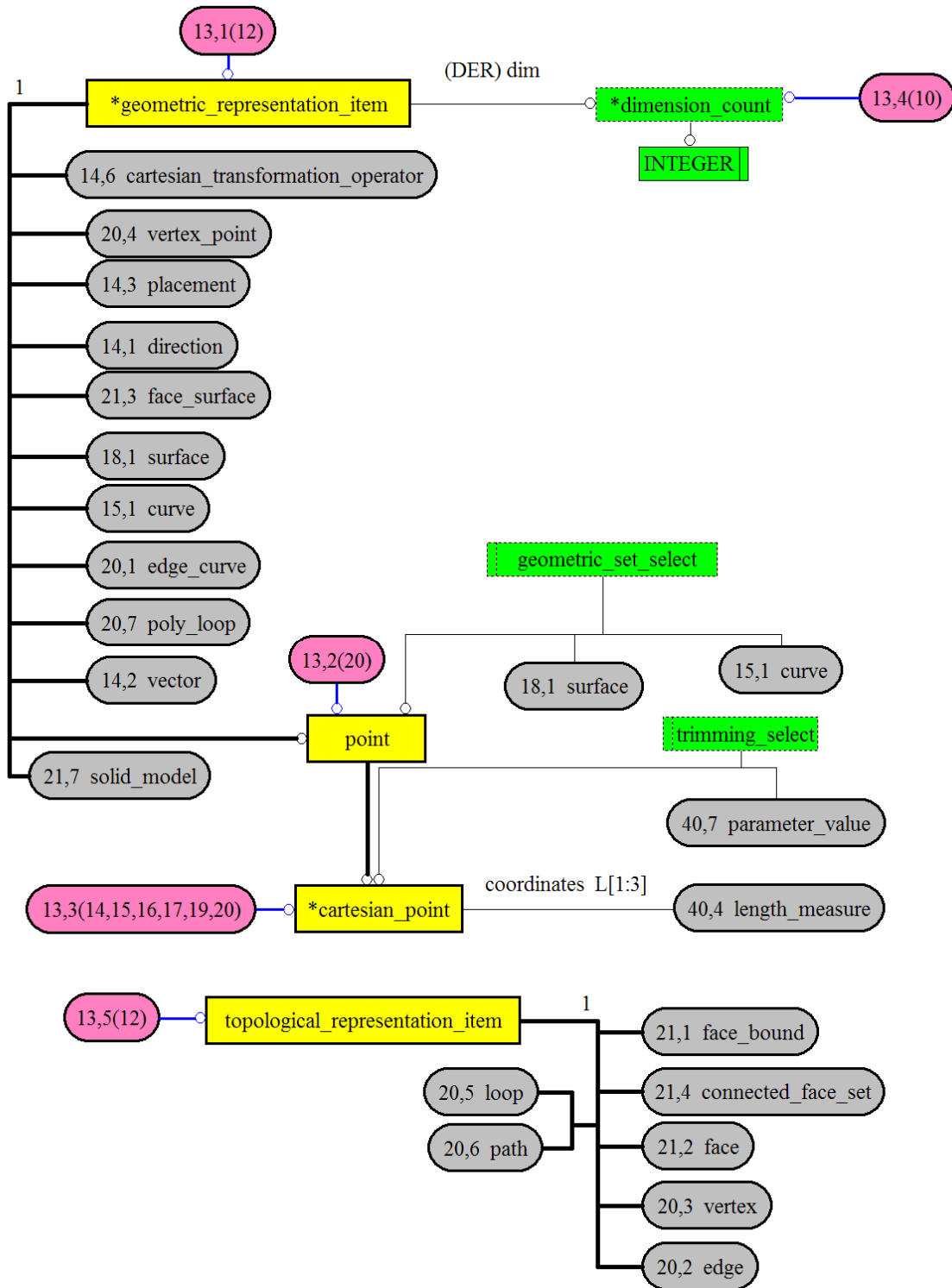


Figure H.13 — AIM EXPRESS-G diagram point

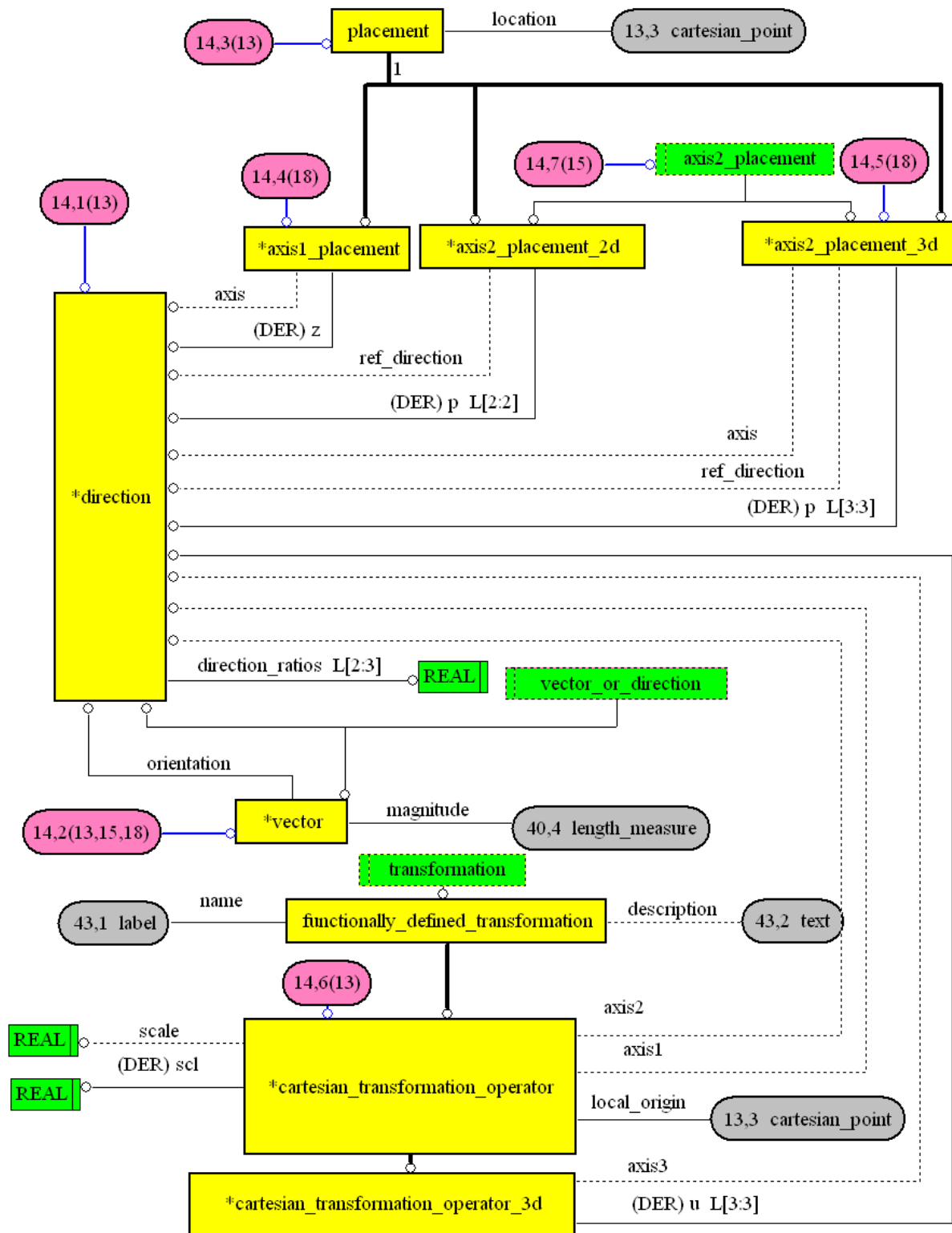


Figure H.14 — AIM EXPRESS-G diagram placement

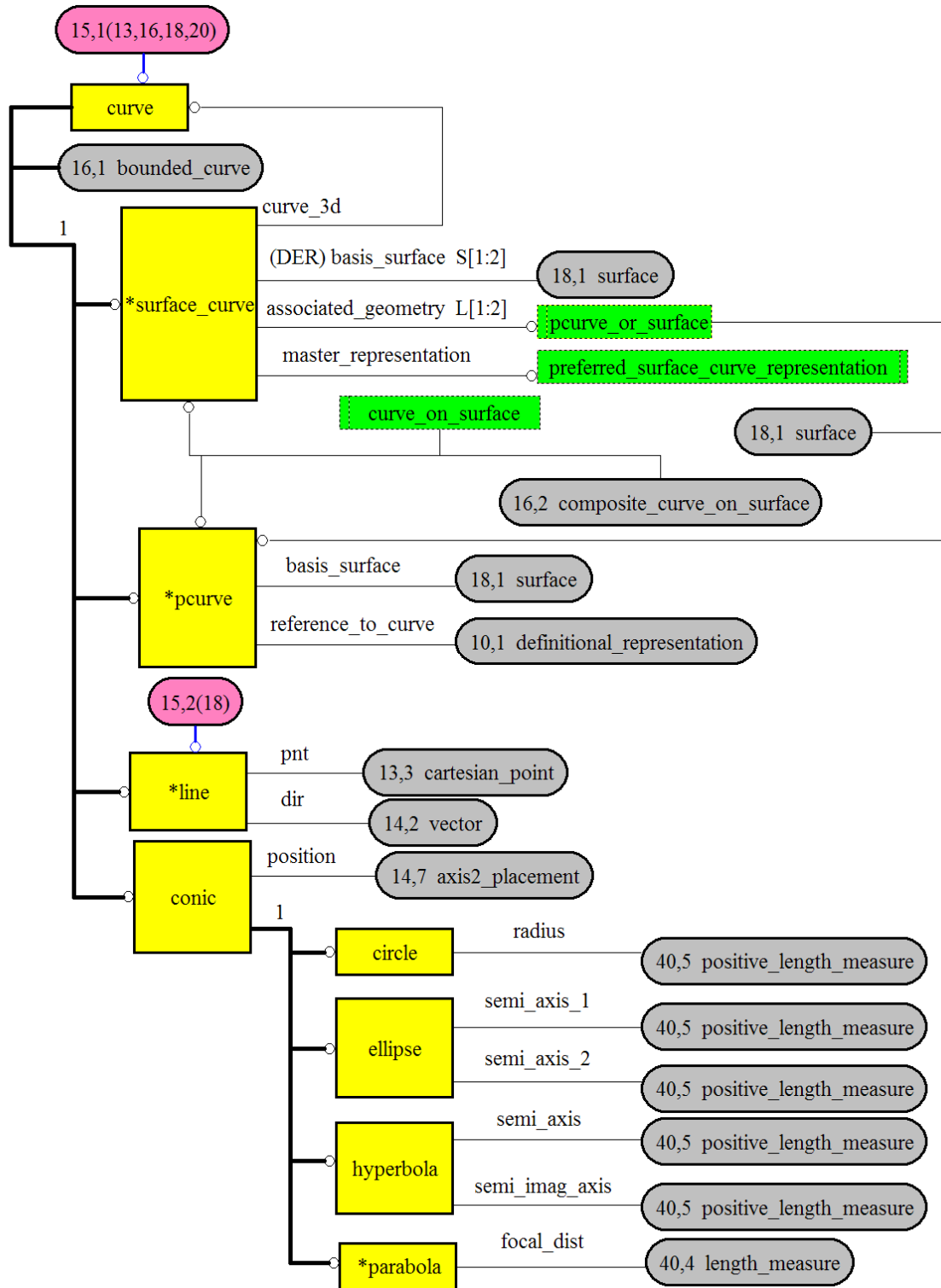


Figure H.15 — AIM EXPRESS-G diagram curve

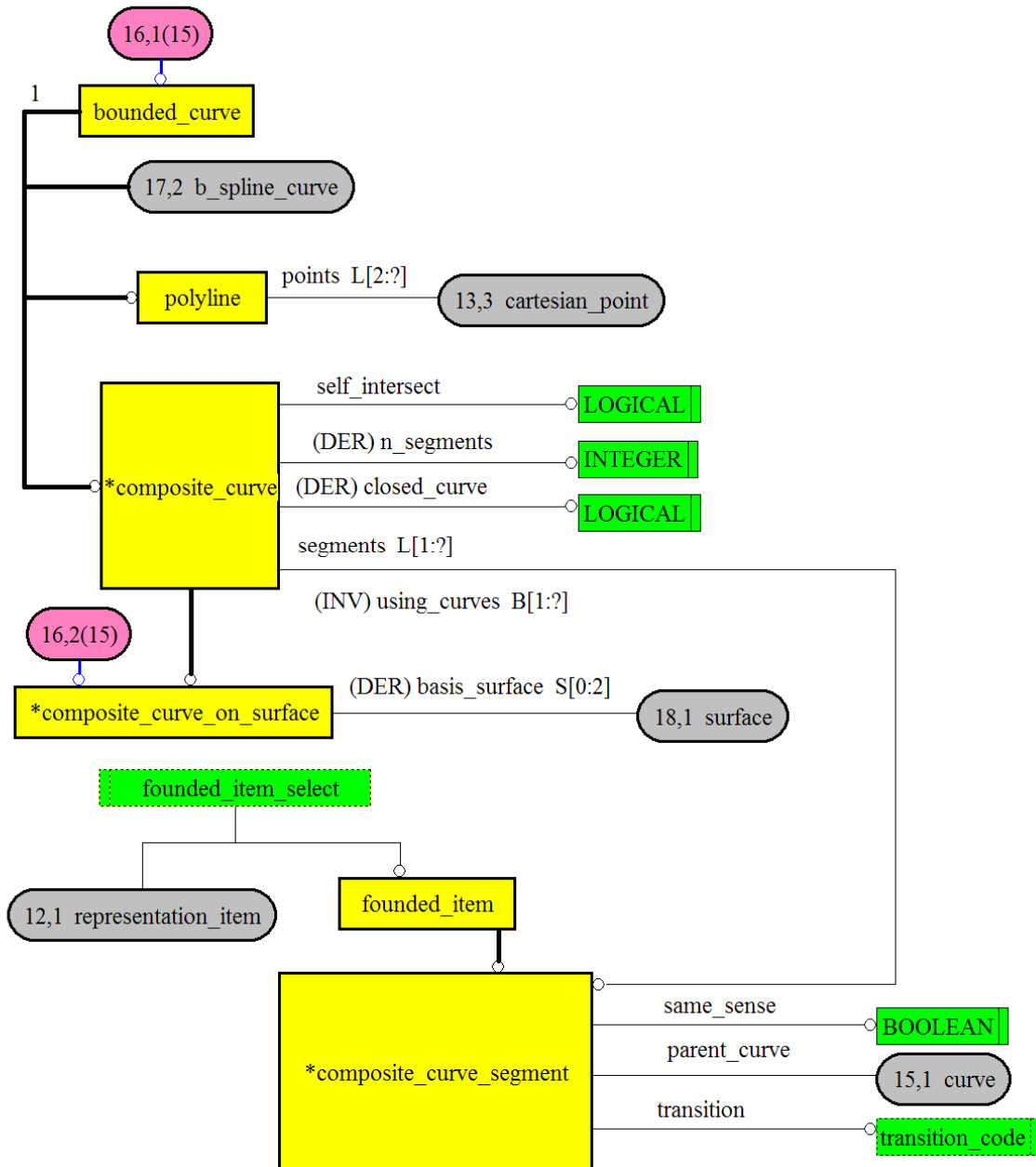
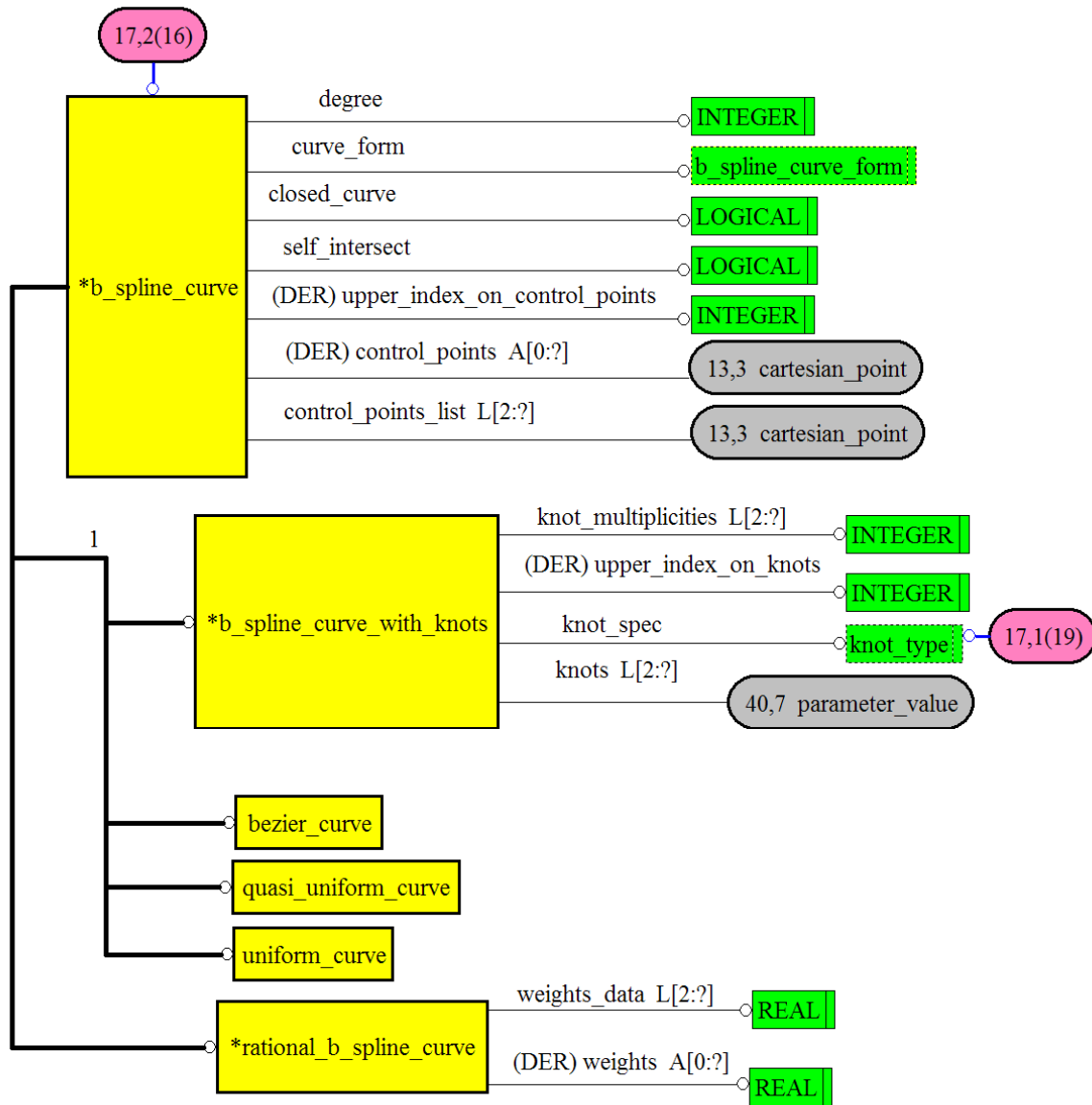


Figure H.16 — AIM EXPRESS-G diagram bounded\_curve



**Figure H.17 — AIM EXPRESS-G diagram b-spline\_curve**

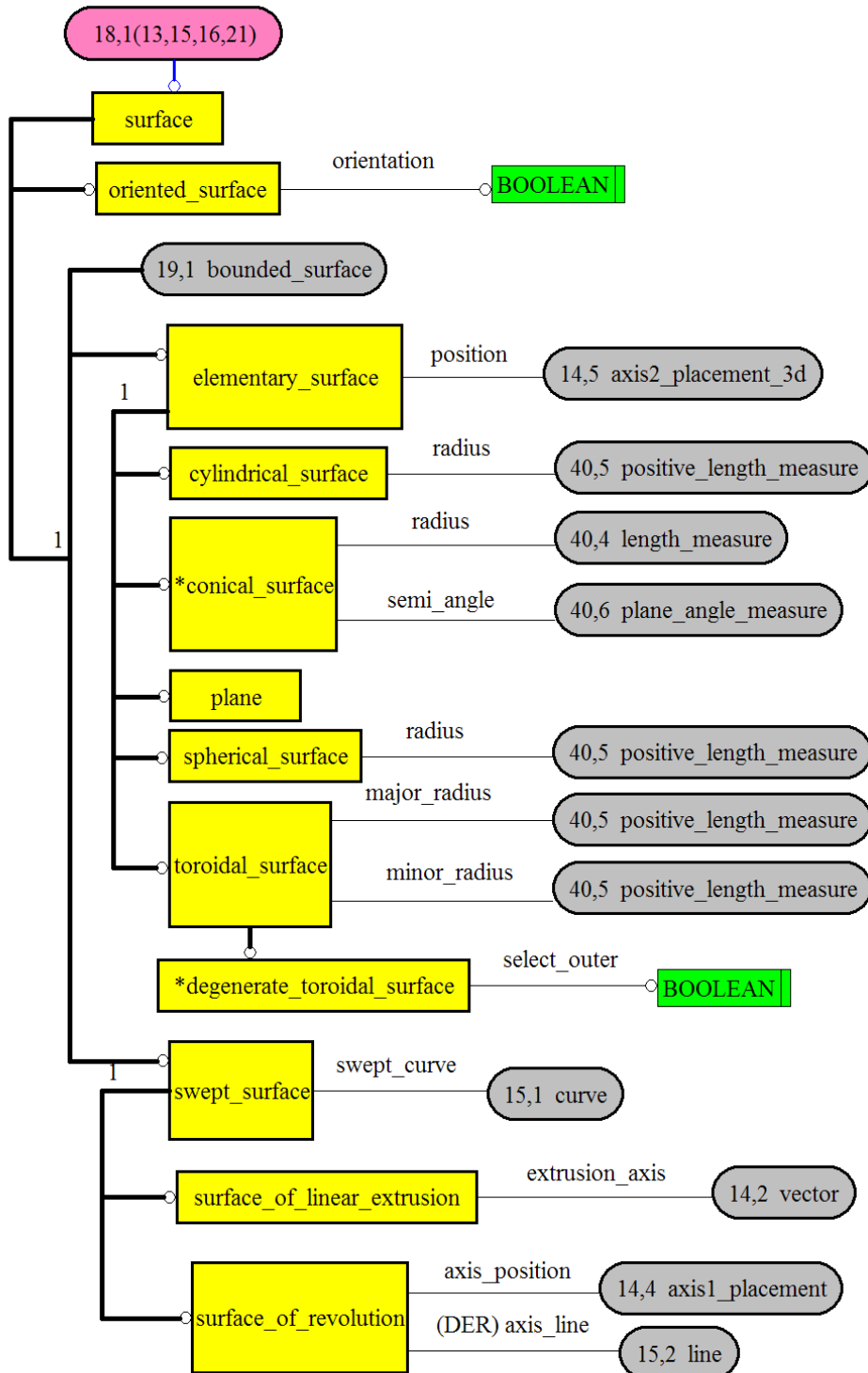


Figure H.18 — AIM EXPRESS-G diagram surface



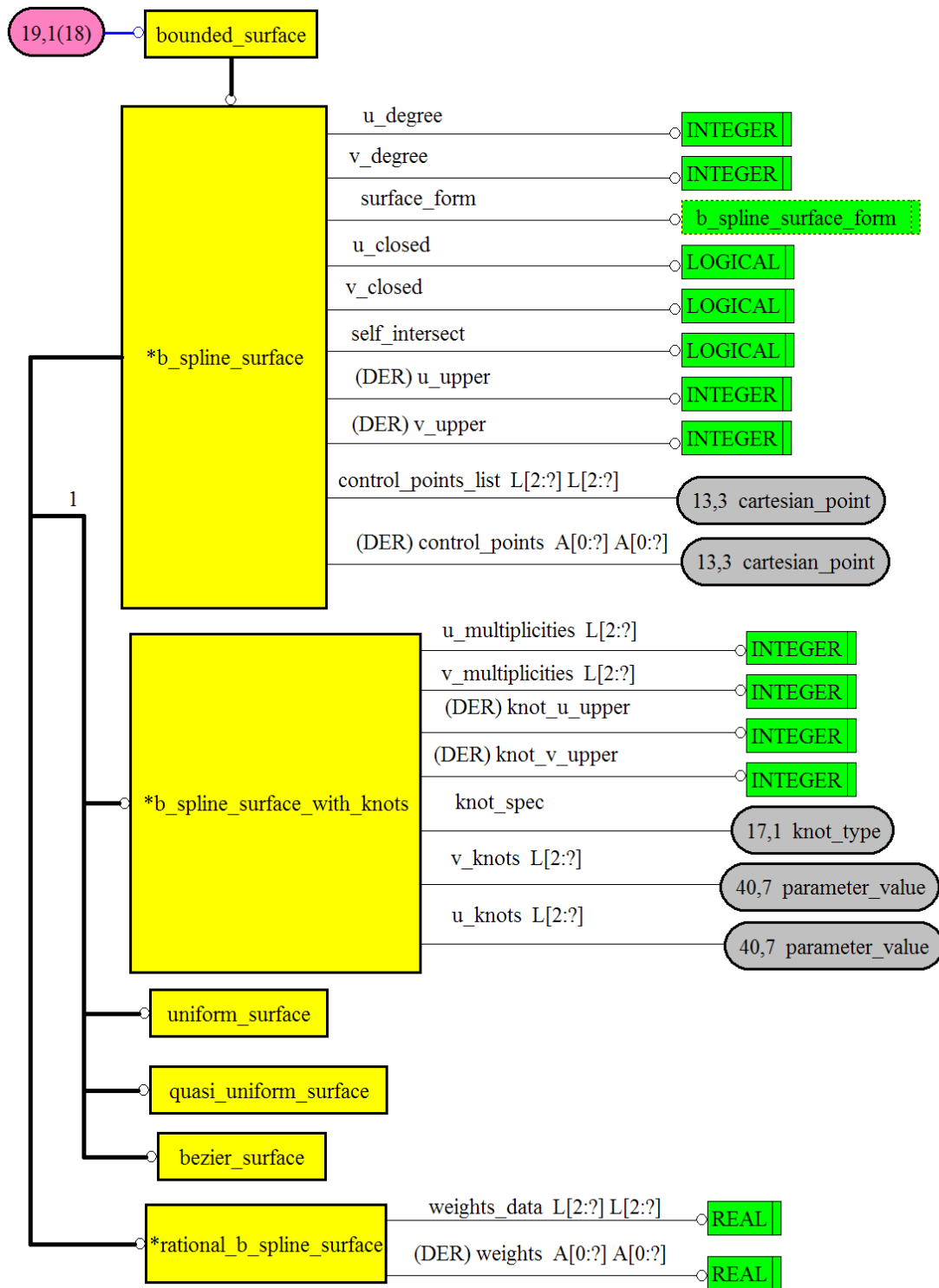


Figure H.19 — AIM EXPRESS-G diagram bounded\_surface

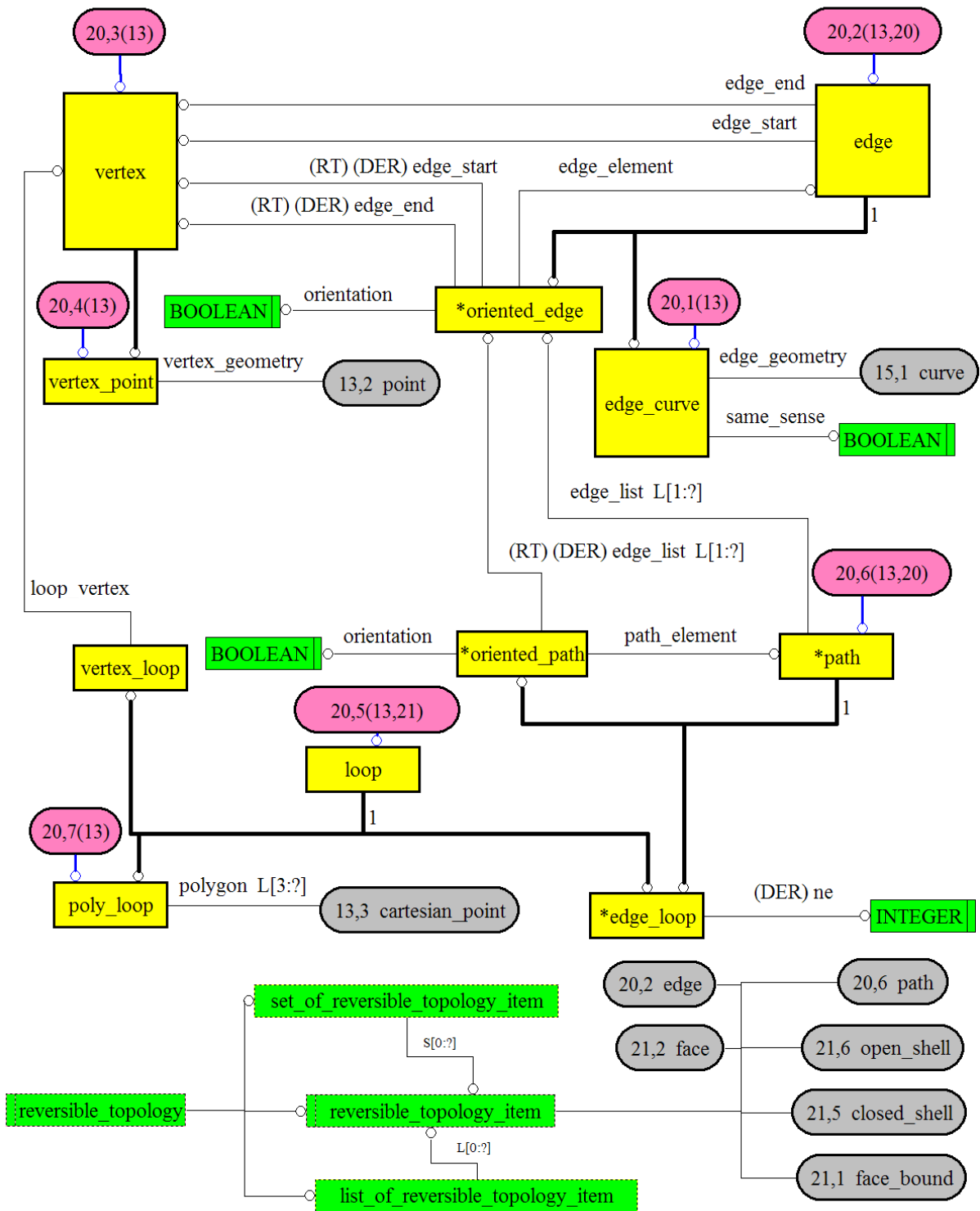


Figure H.20 — AIM EXPRESS-G diagram topology

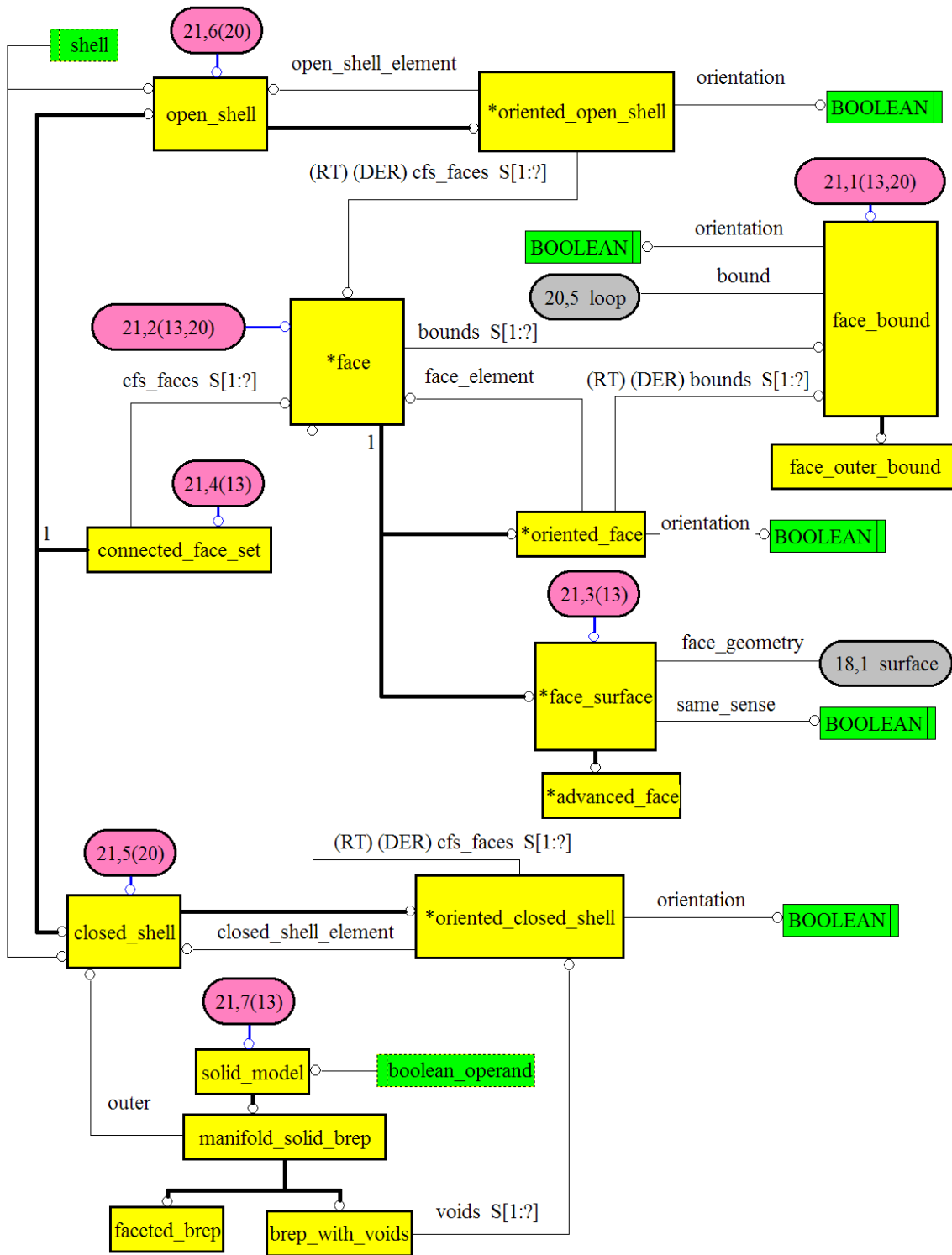


Figure H.21 — AIM EXPRESS-G diagram shell and face\_bound

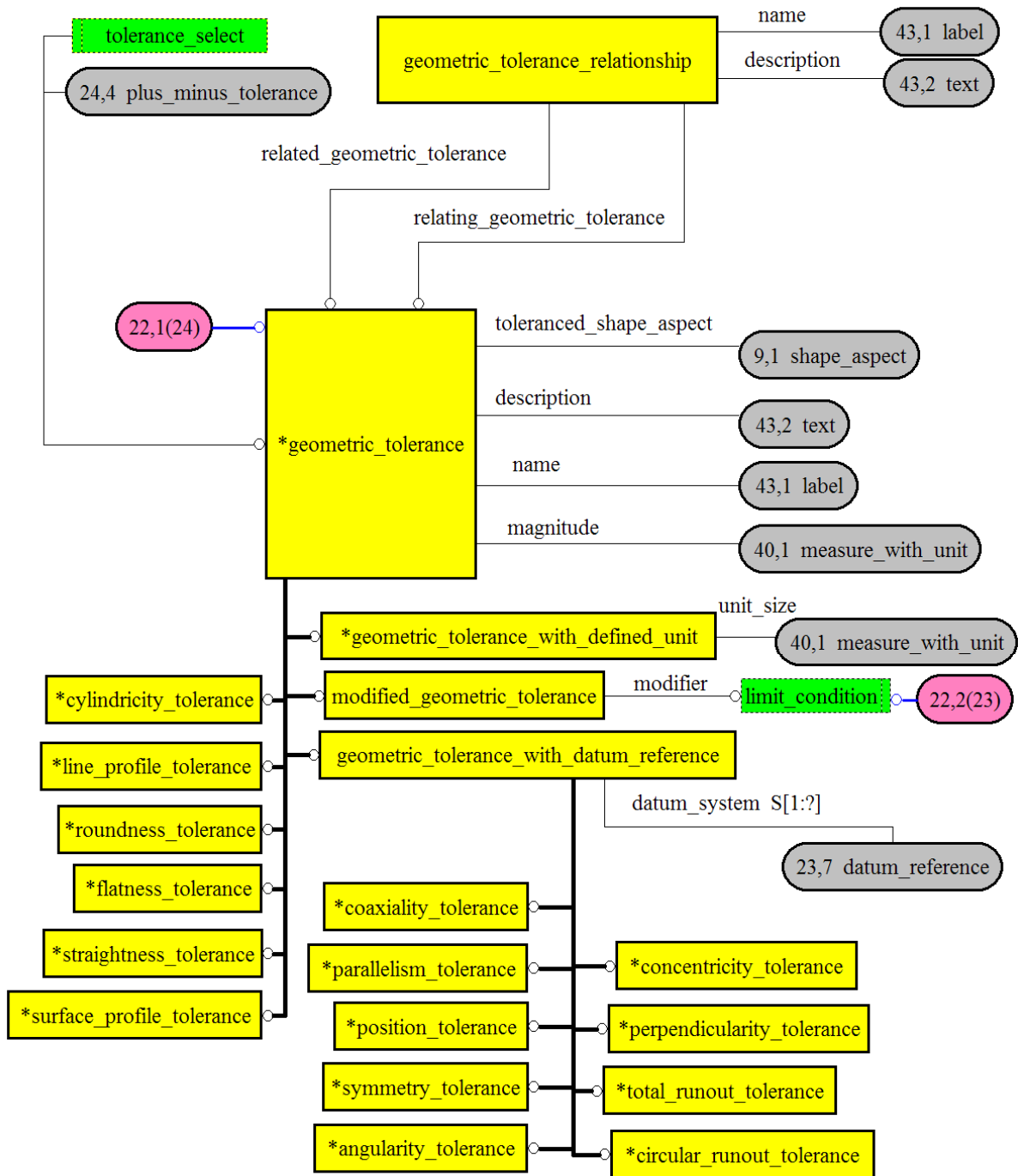


Figure H.22 — AIM EXPRESS-G diagram geometric\_tolerance

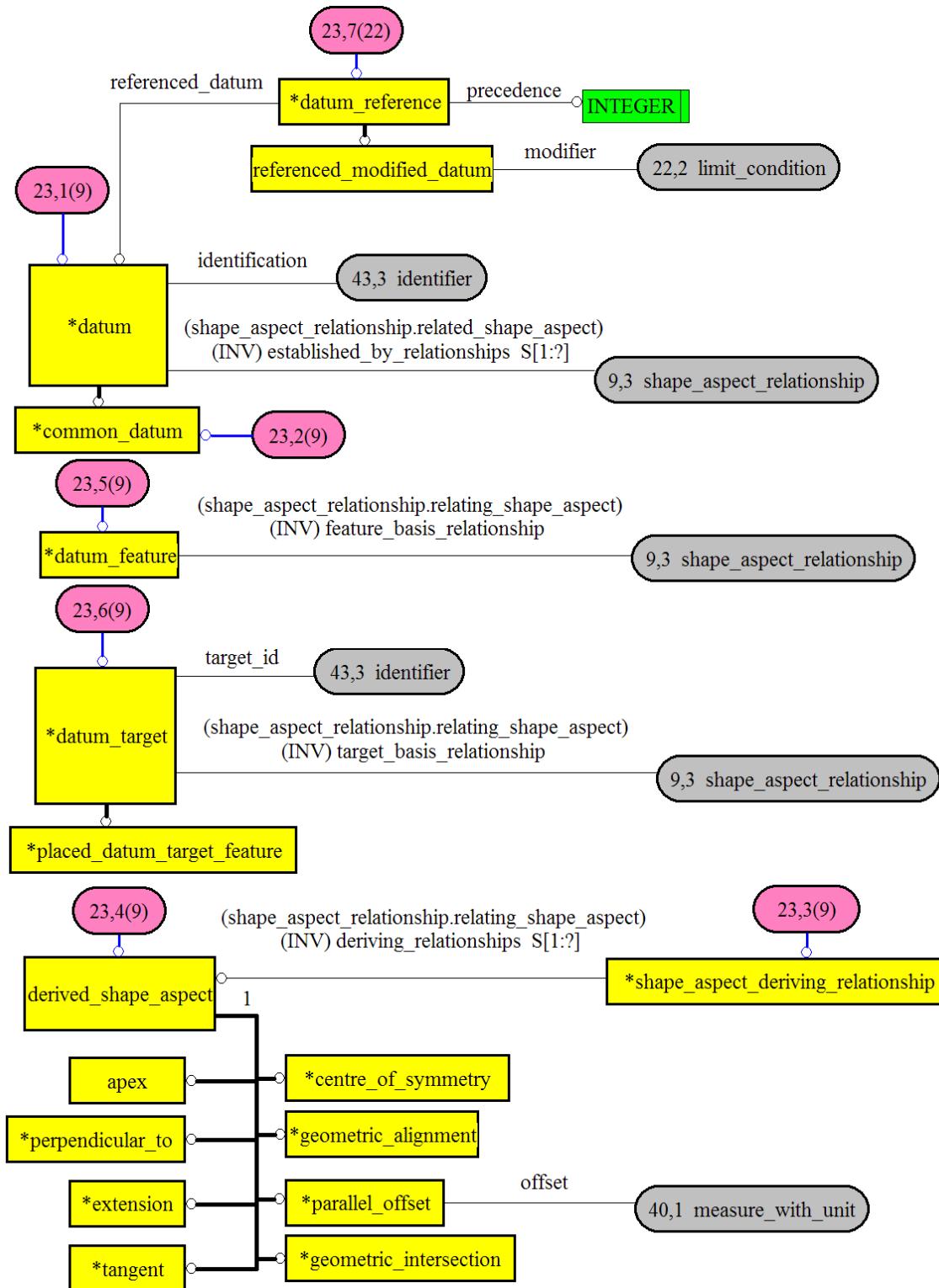


Figure H.23 — AIM EXPRESS-G diagram datum

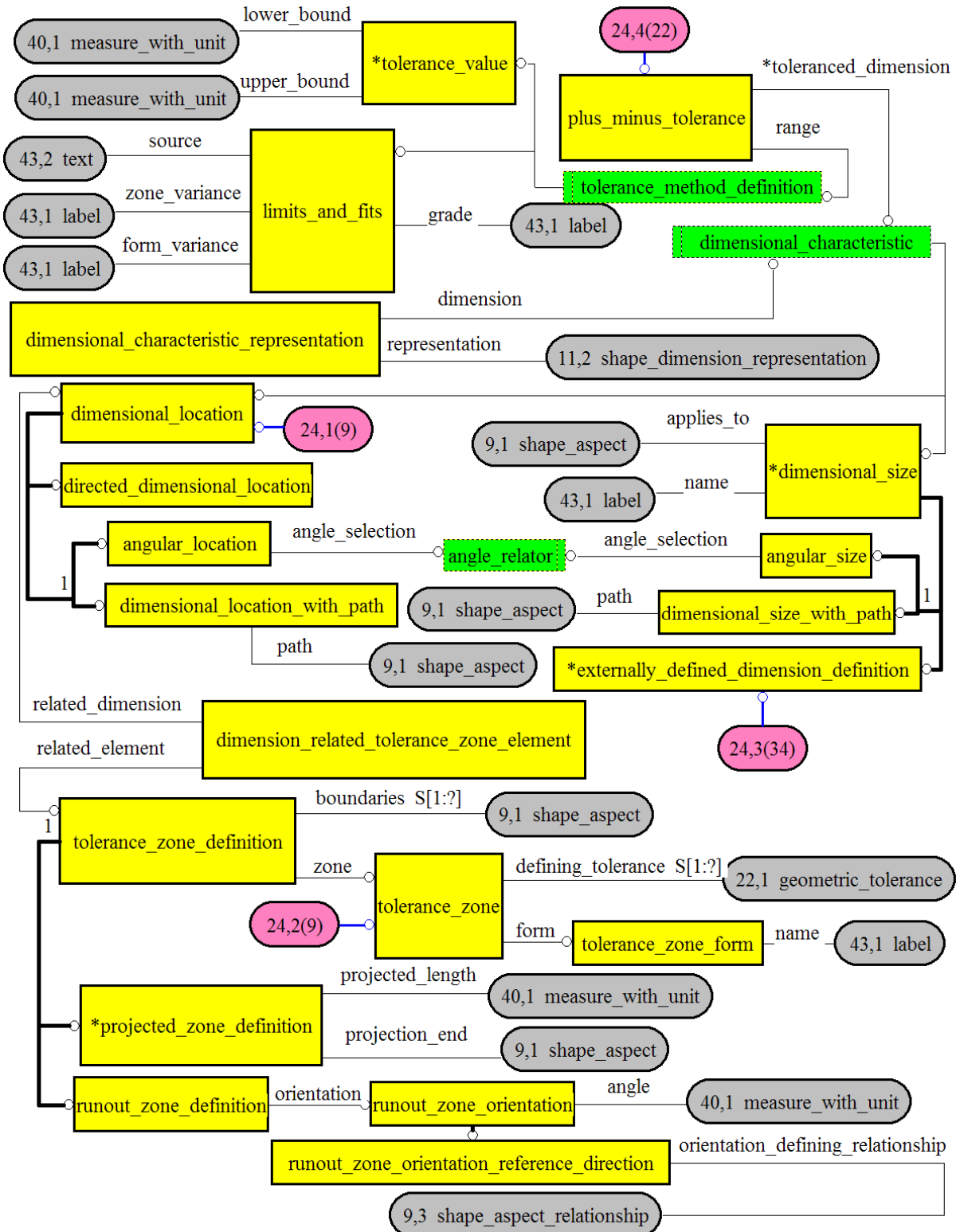
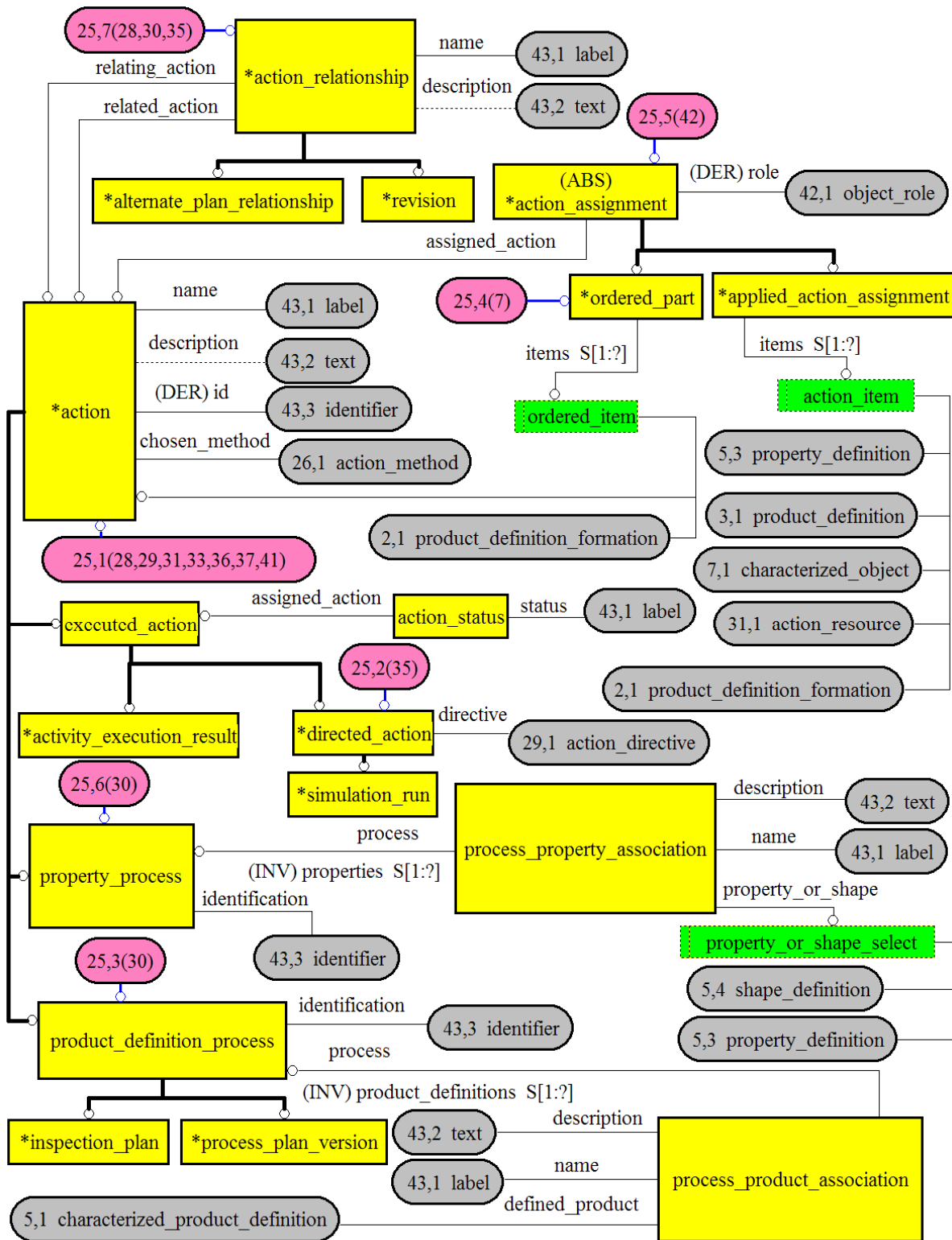


Figure H.24 — AIM EXPRESS-G diagram dimensional tolerance



**Figure H.25 — AIM EXPRESS-G diagram action**

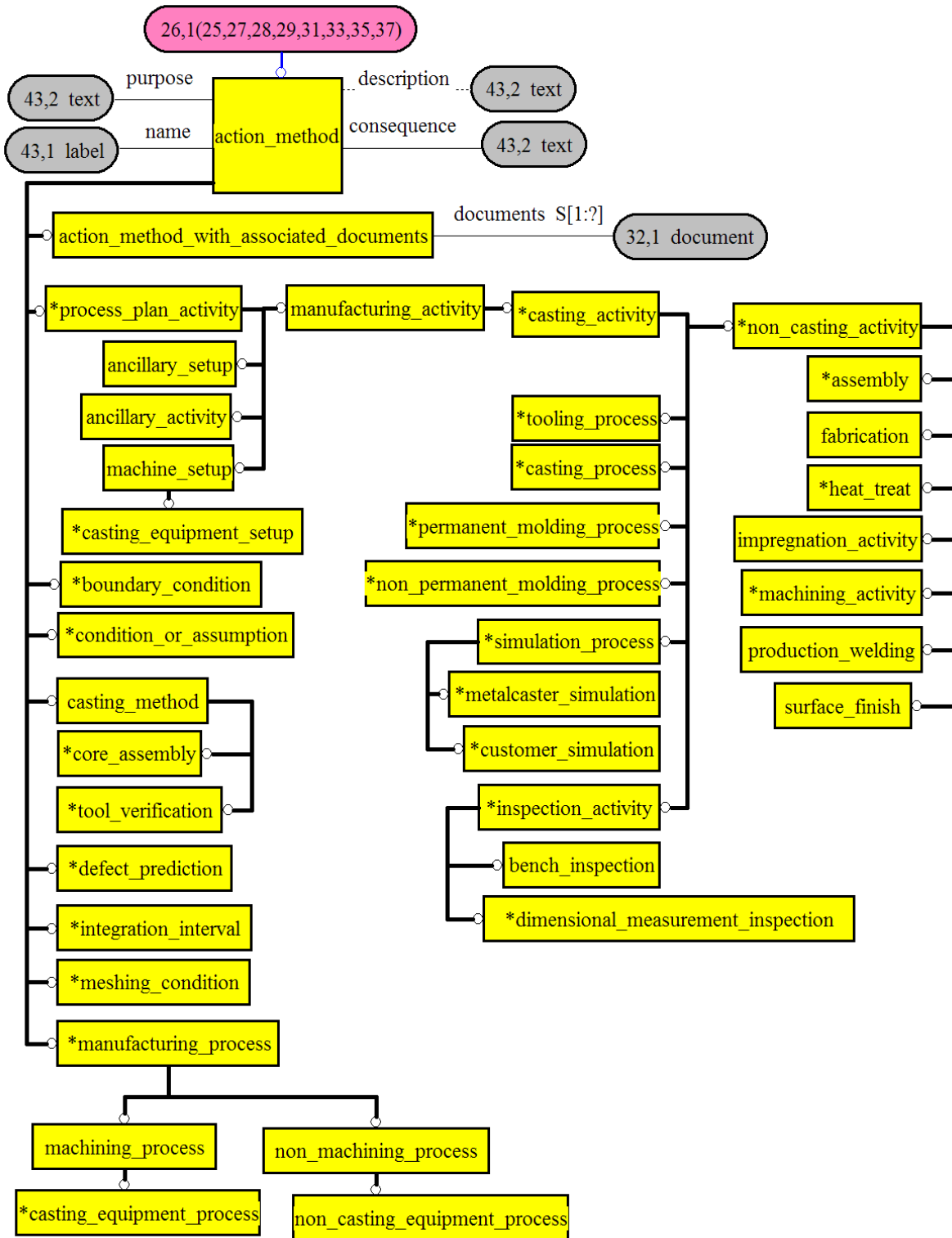
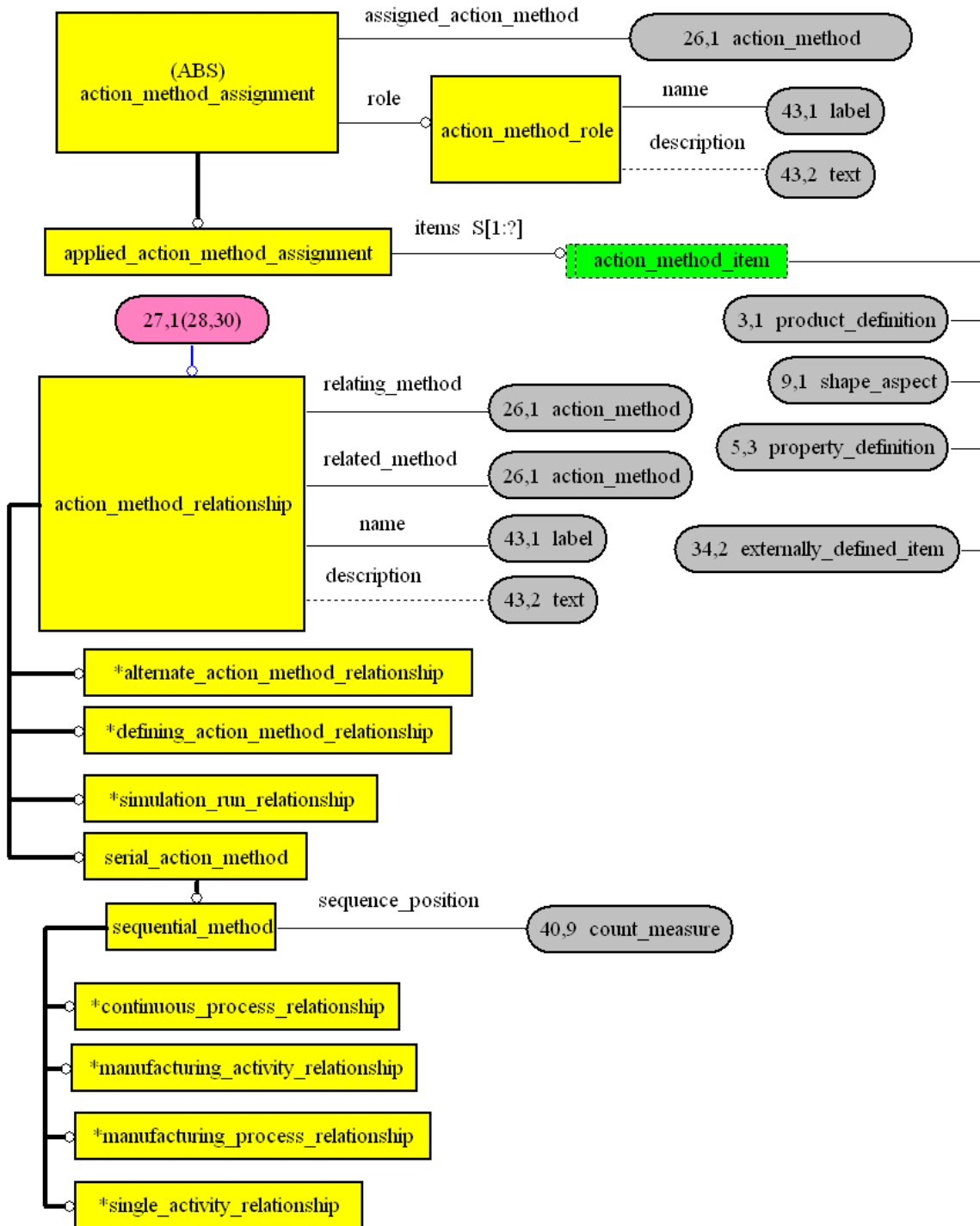


Figure H.26 — AIM EXPRESS-G diagram action\_method





**Figure H.27 — AIM EXPRESS-G diagram action\_method\_relationship**

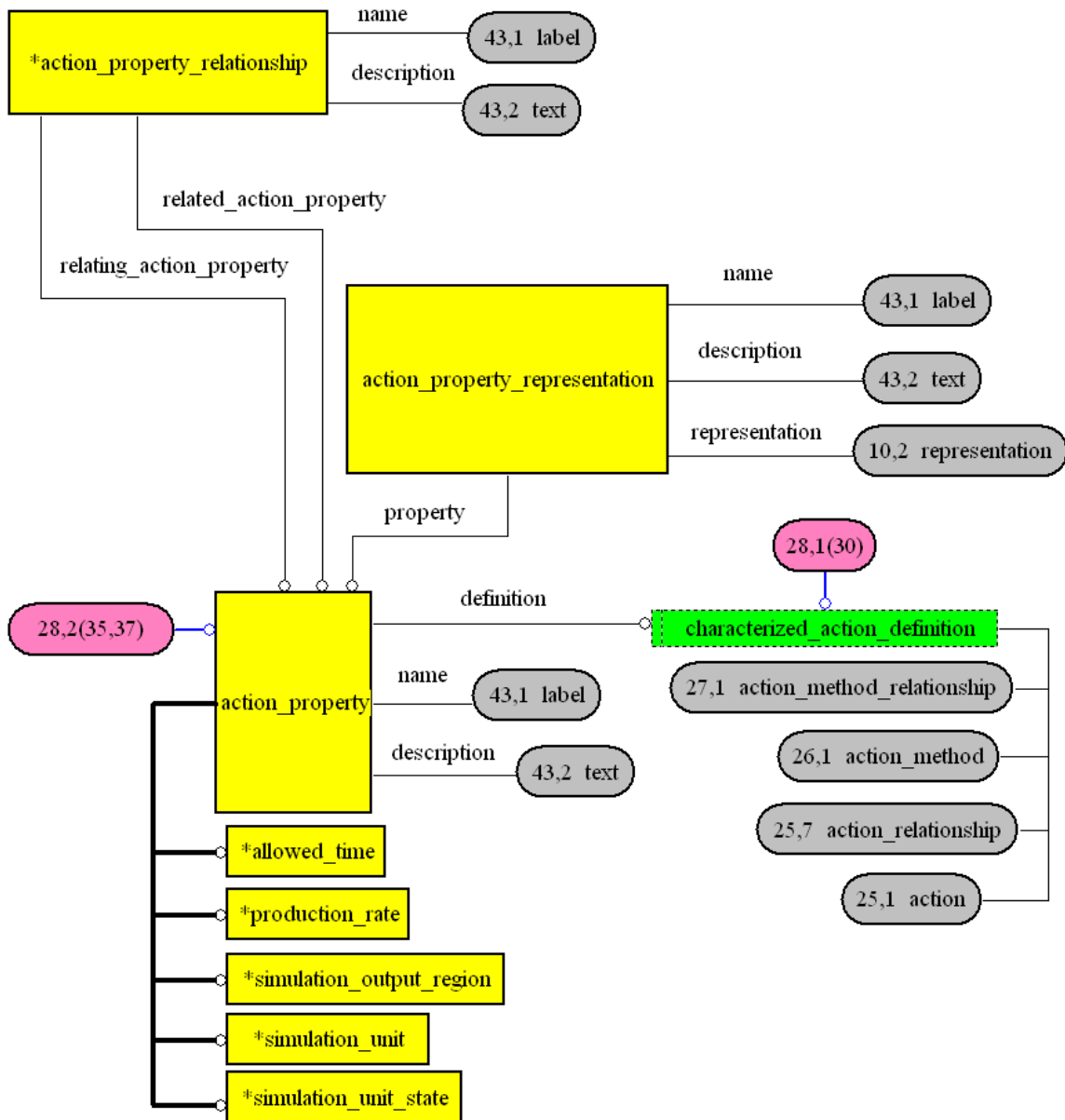
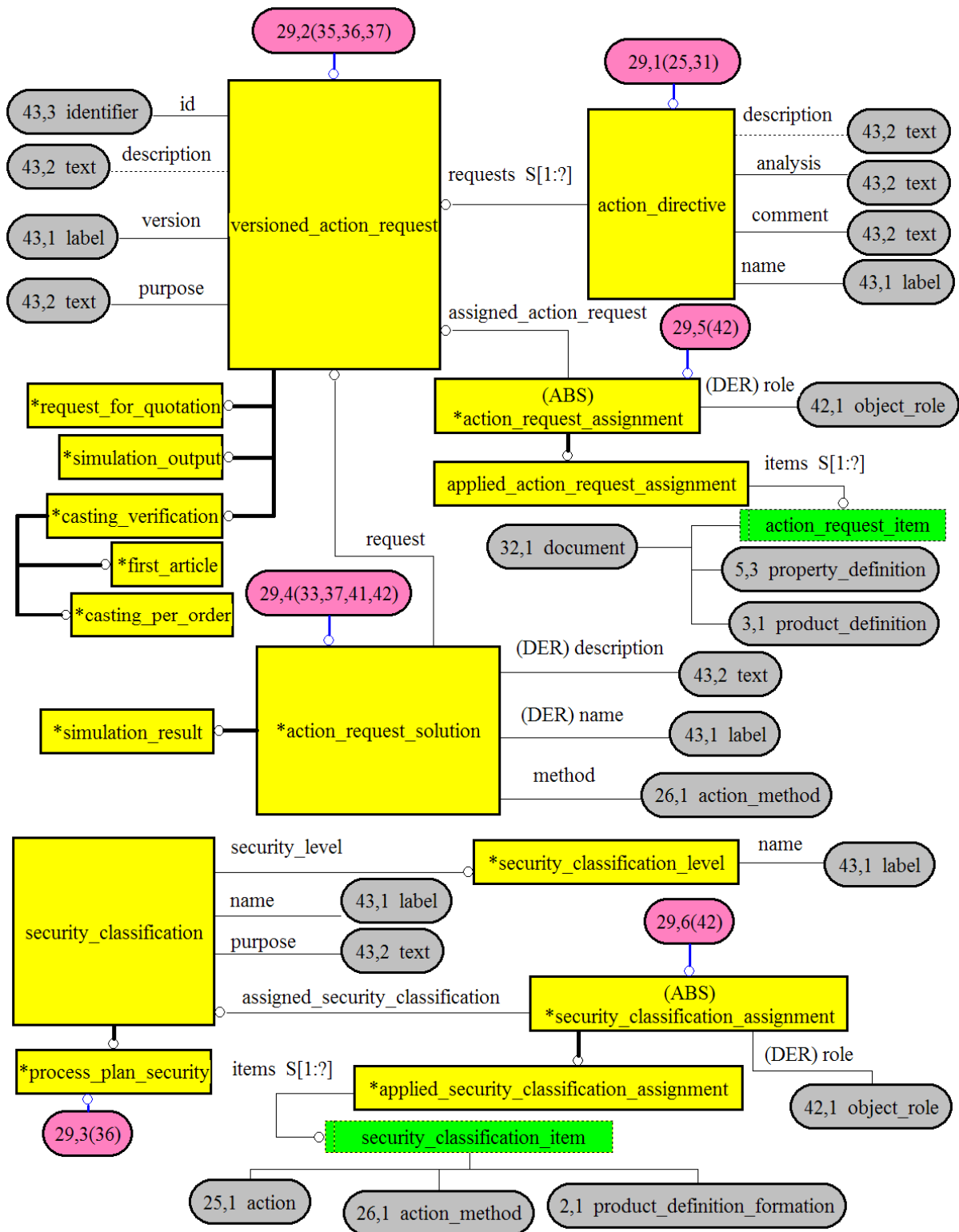
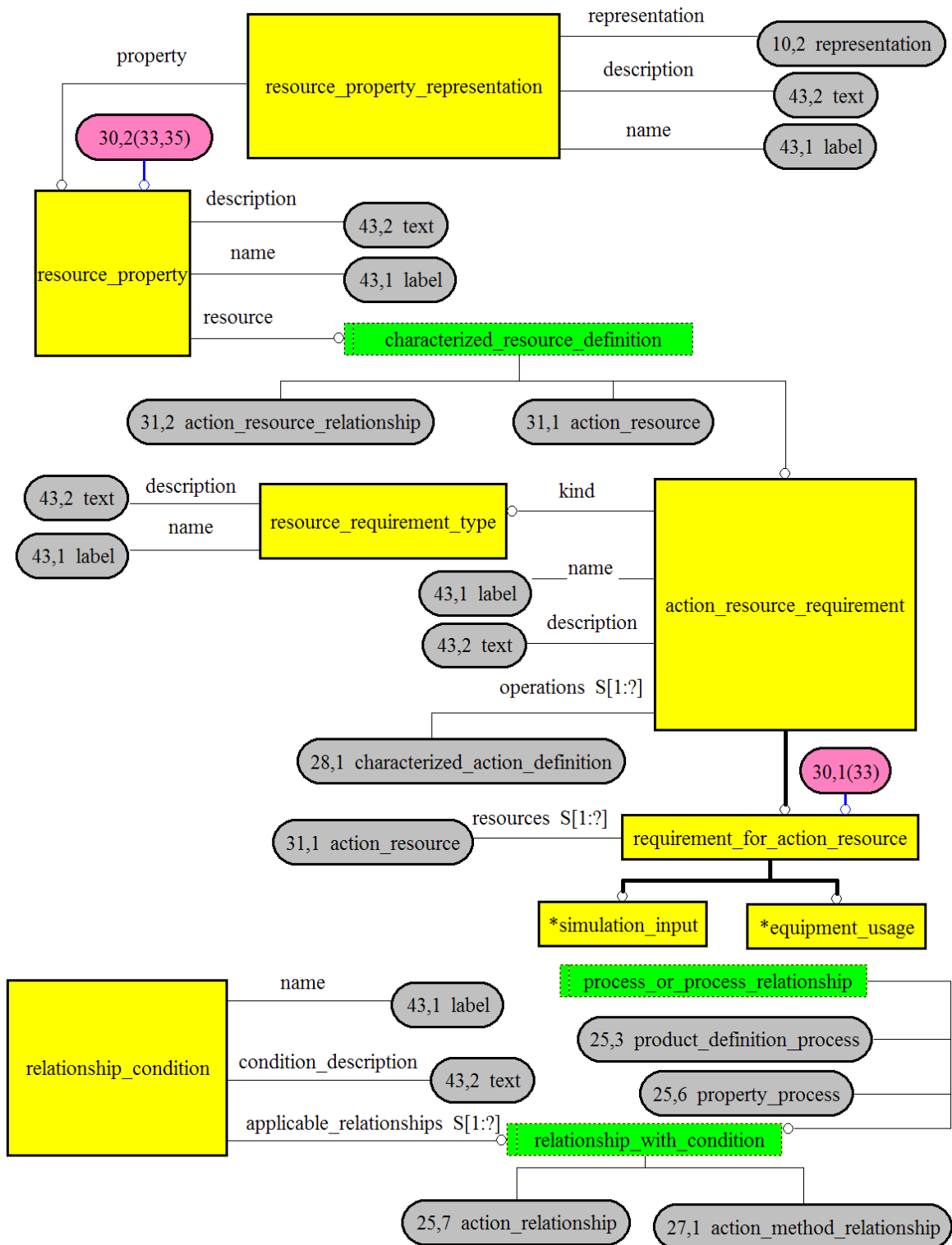


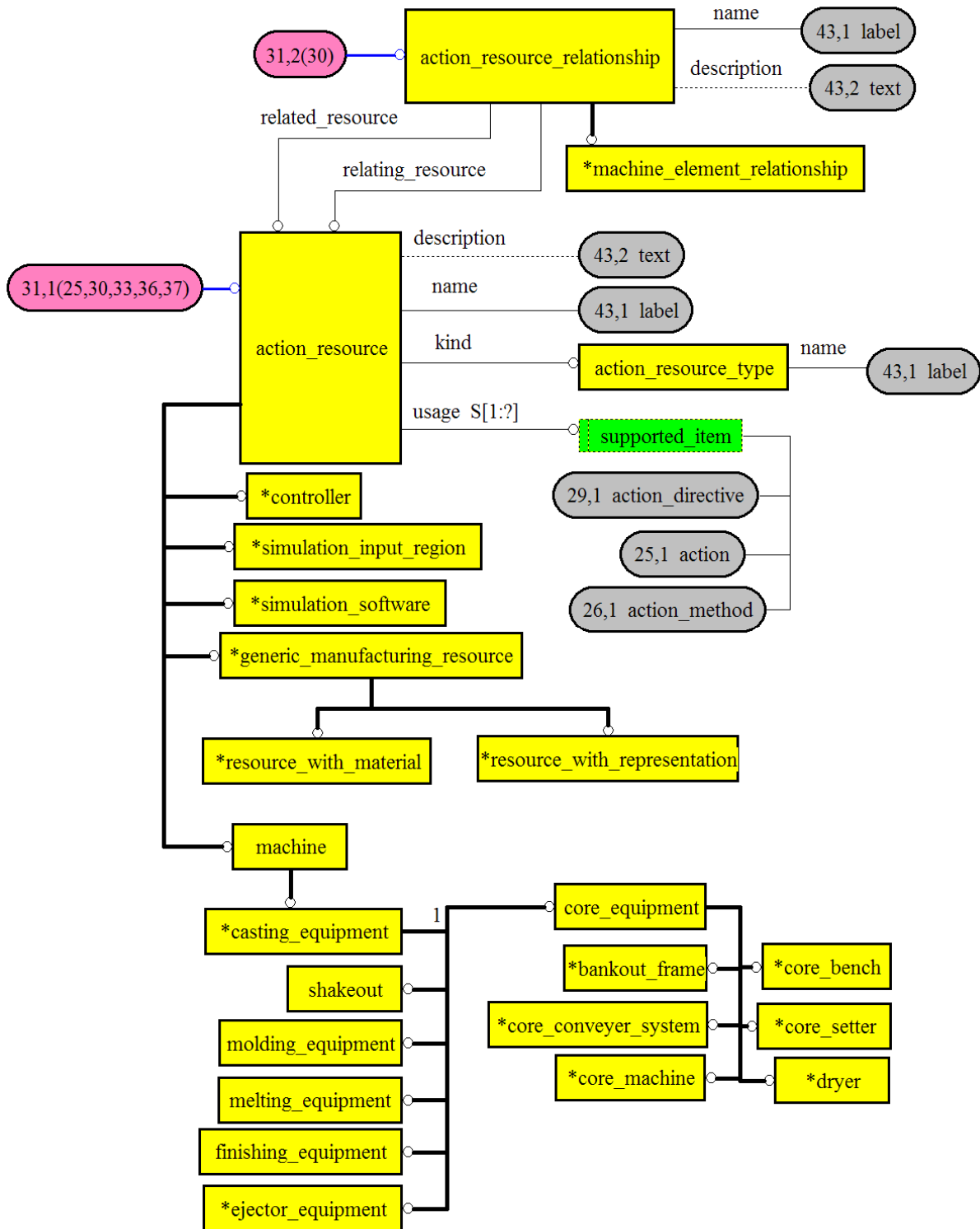
Figure H.28 — AIM EXPRESS-G diagram action\_property



**Figure H.29 — AIM EXPRESS-G diagram action\_directive**



**Figure H.30 — AIM EXPRESS-G diagram resource\_property**



**Figure H.31 — AIM EXPRESS-G diagram action\_resource**

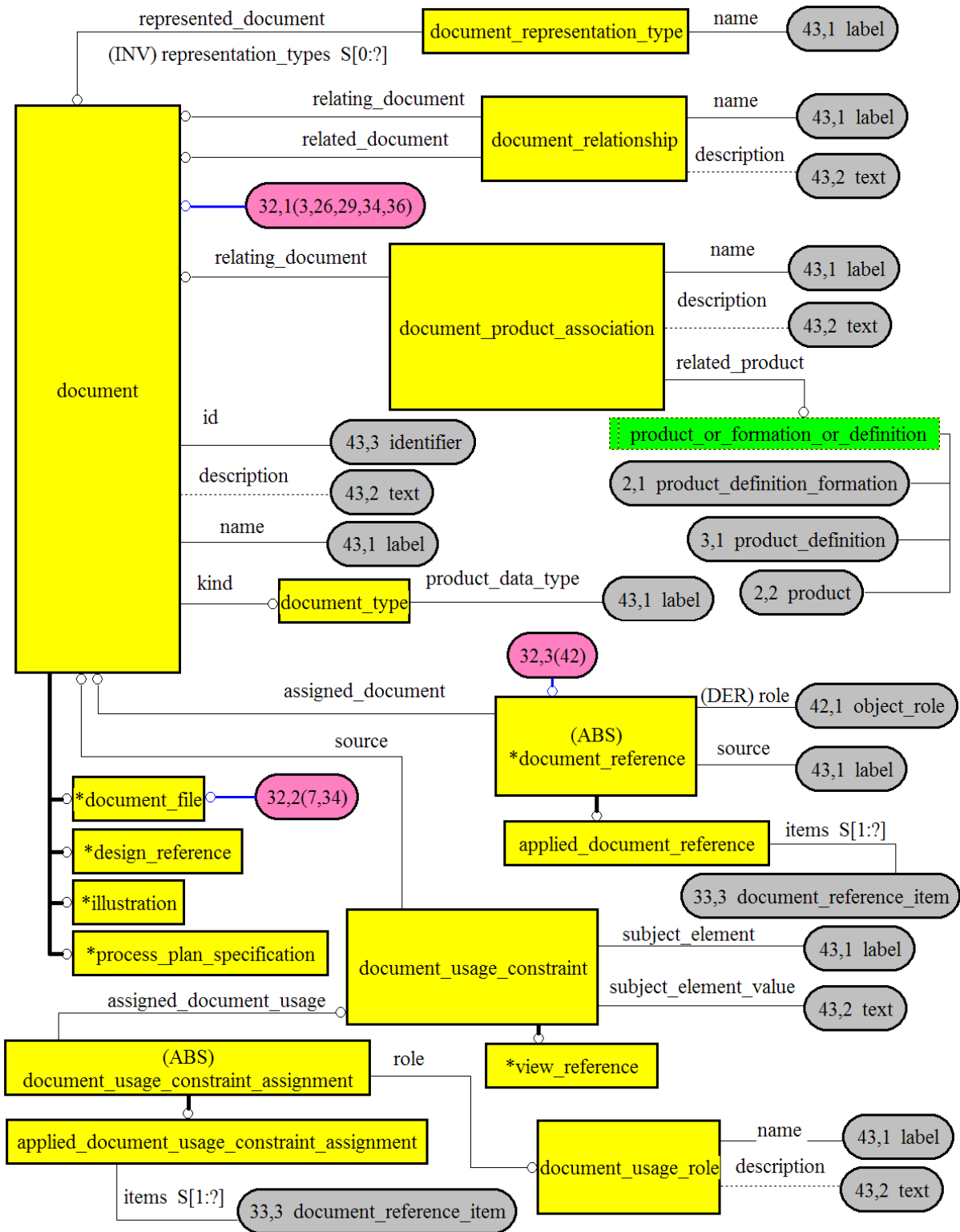


Figure H.32 — AIM EXPRESS-G diagram document

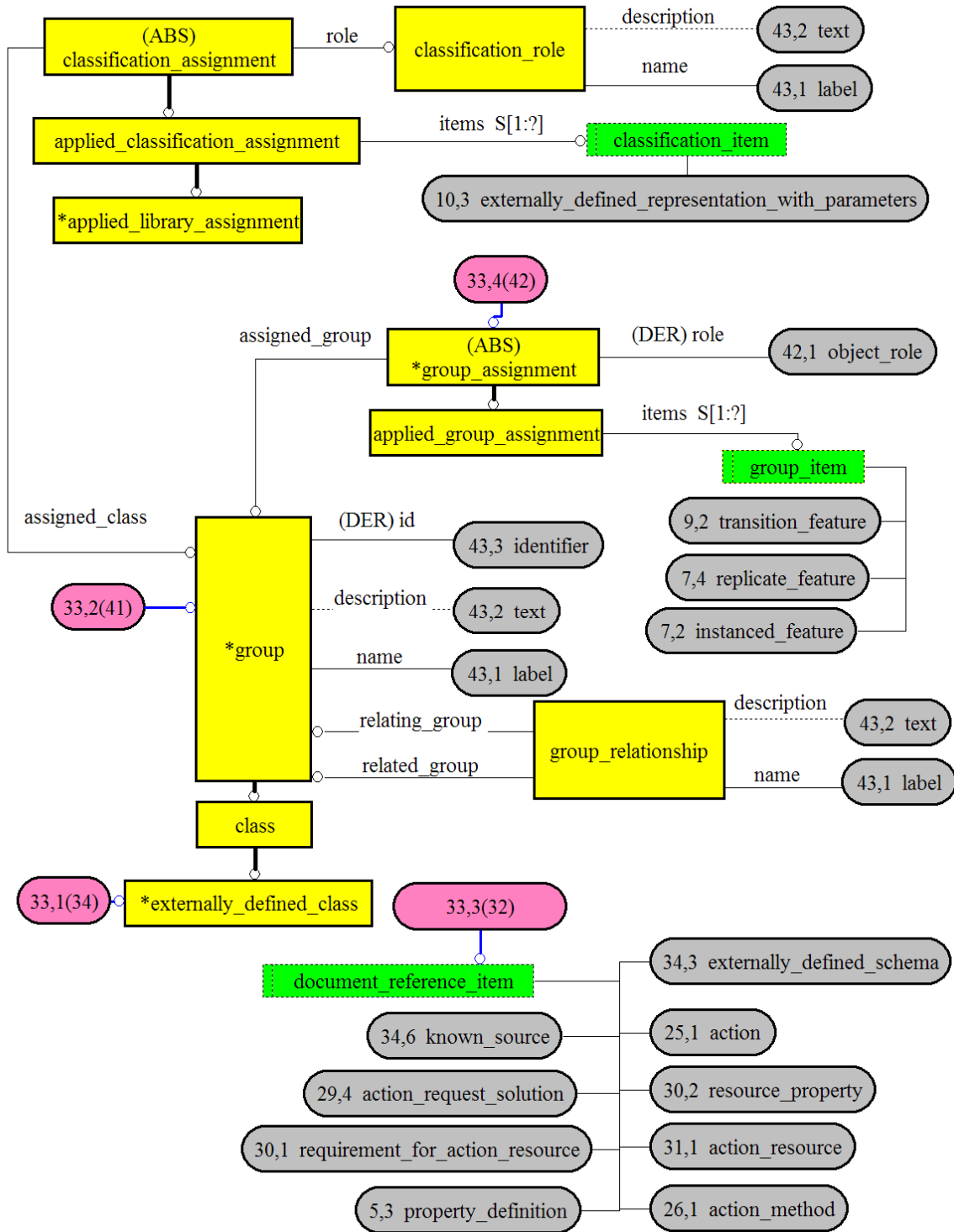


Figure H.33 — AIM EXPRESS-G diagram group

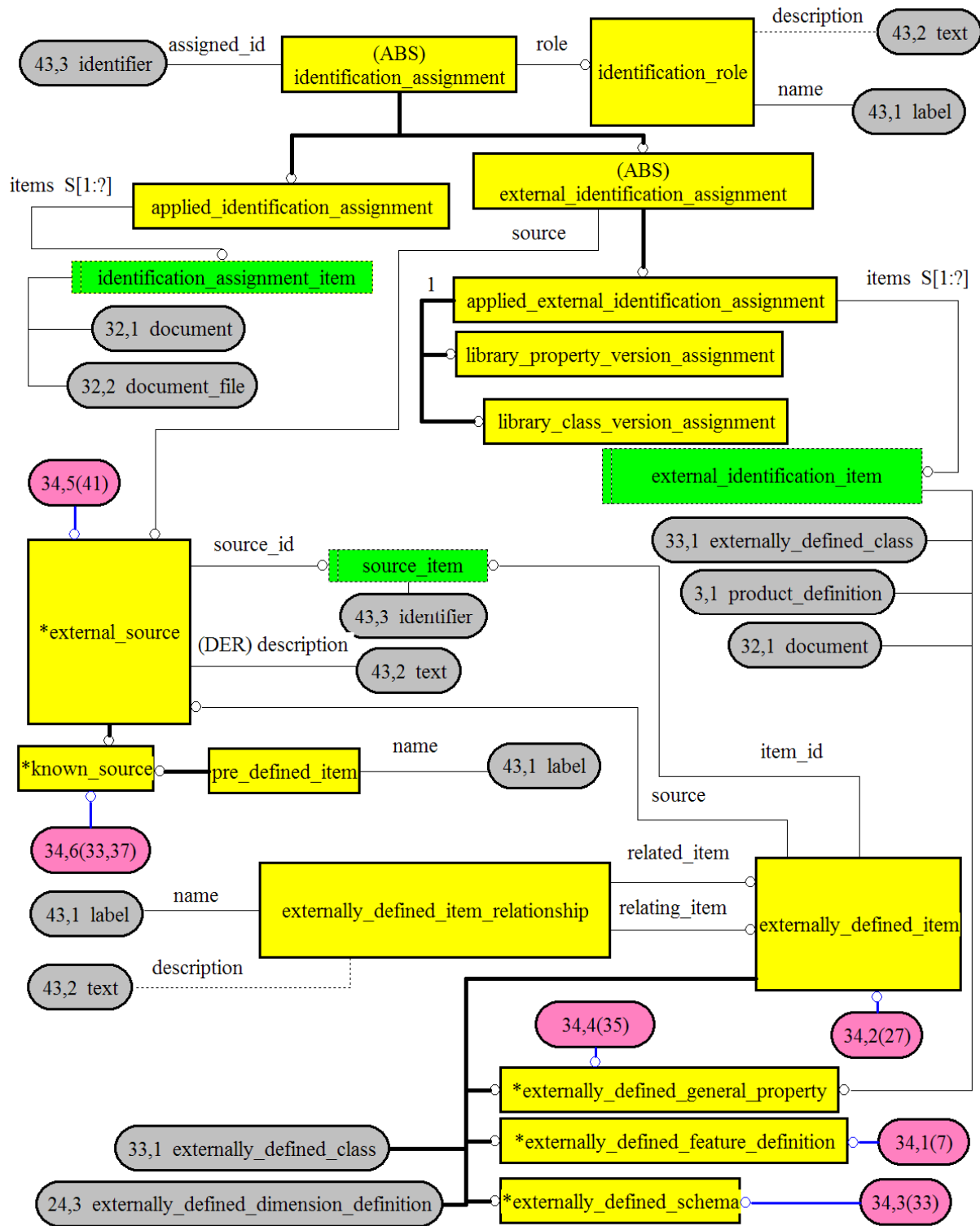


Figure H.34 — AIM EXPRESS-G diagram externally\_defined\_item



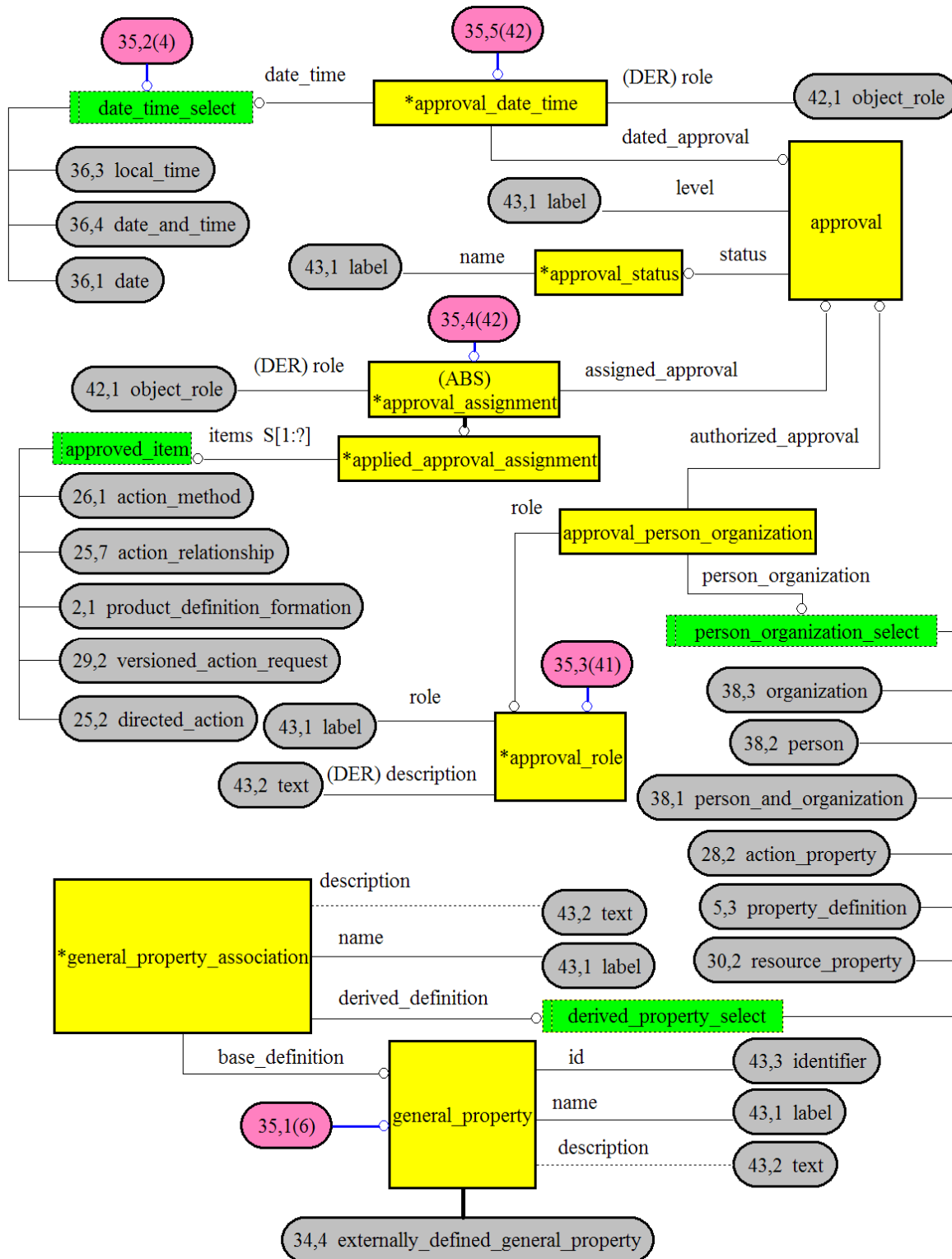


Figure H.35 — AIM EXPRESS-G diagram approval

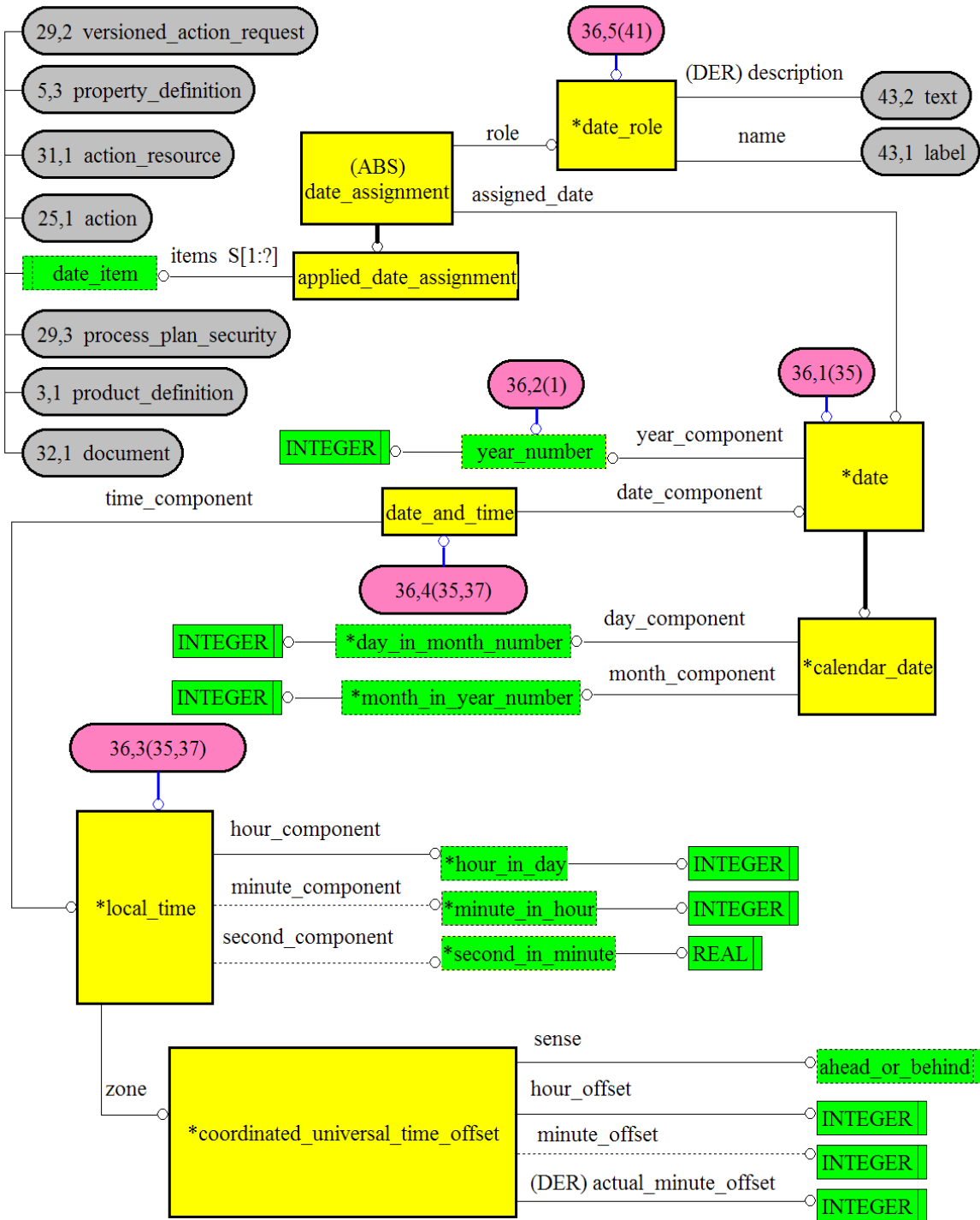


Figure H.36 — AIM EXPRESS-G diagram date

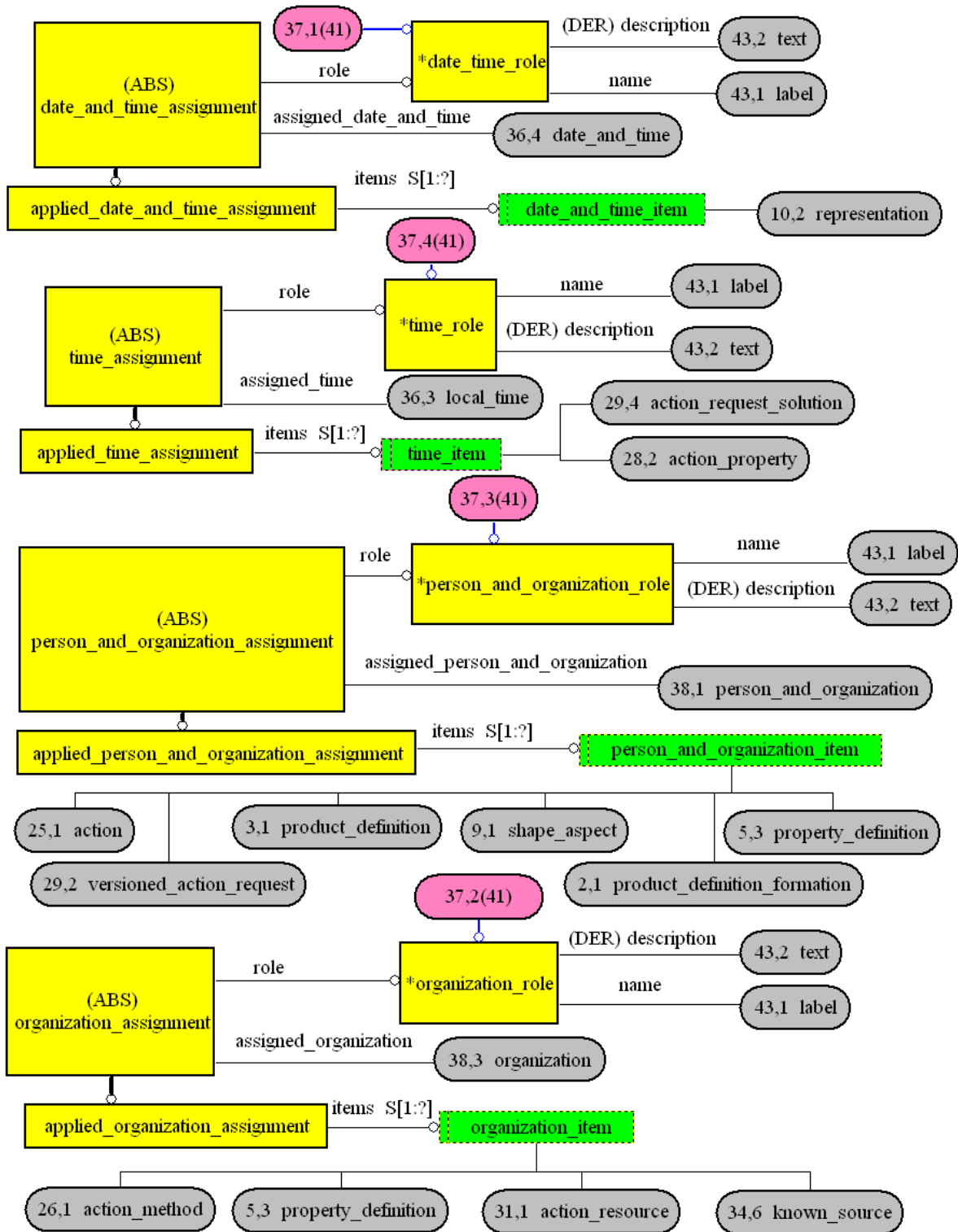


Figure H.37 — AIM EXPRESS-G diagram applied assignments

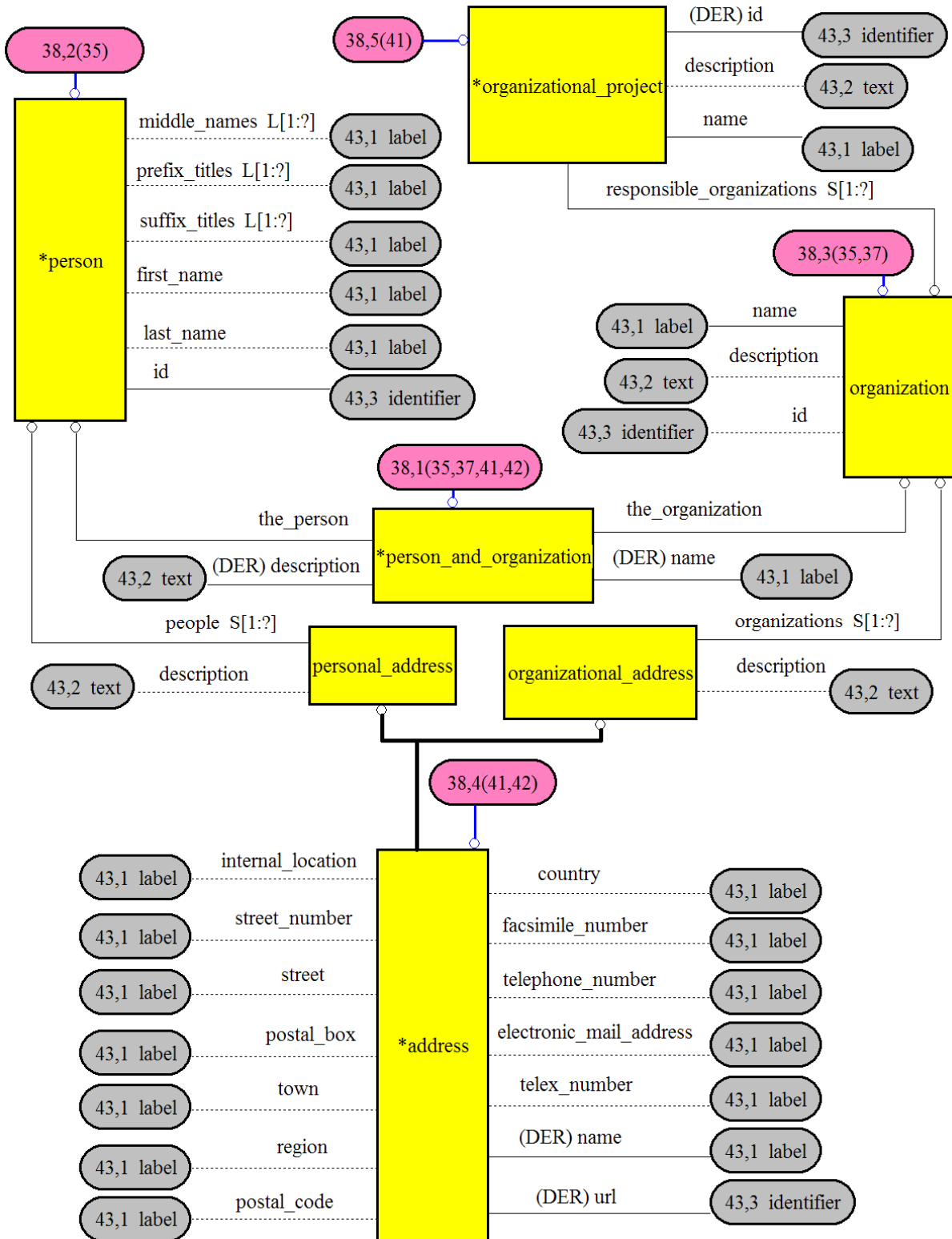


Figure H.38 — AIM EXPRESS-G diagram address

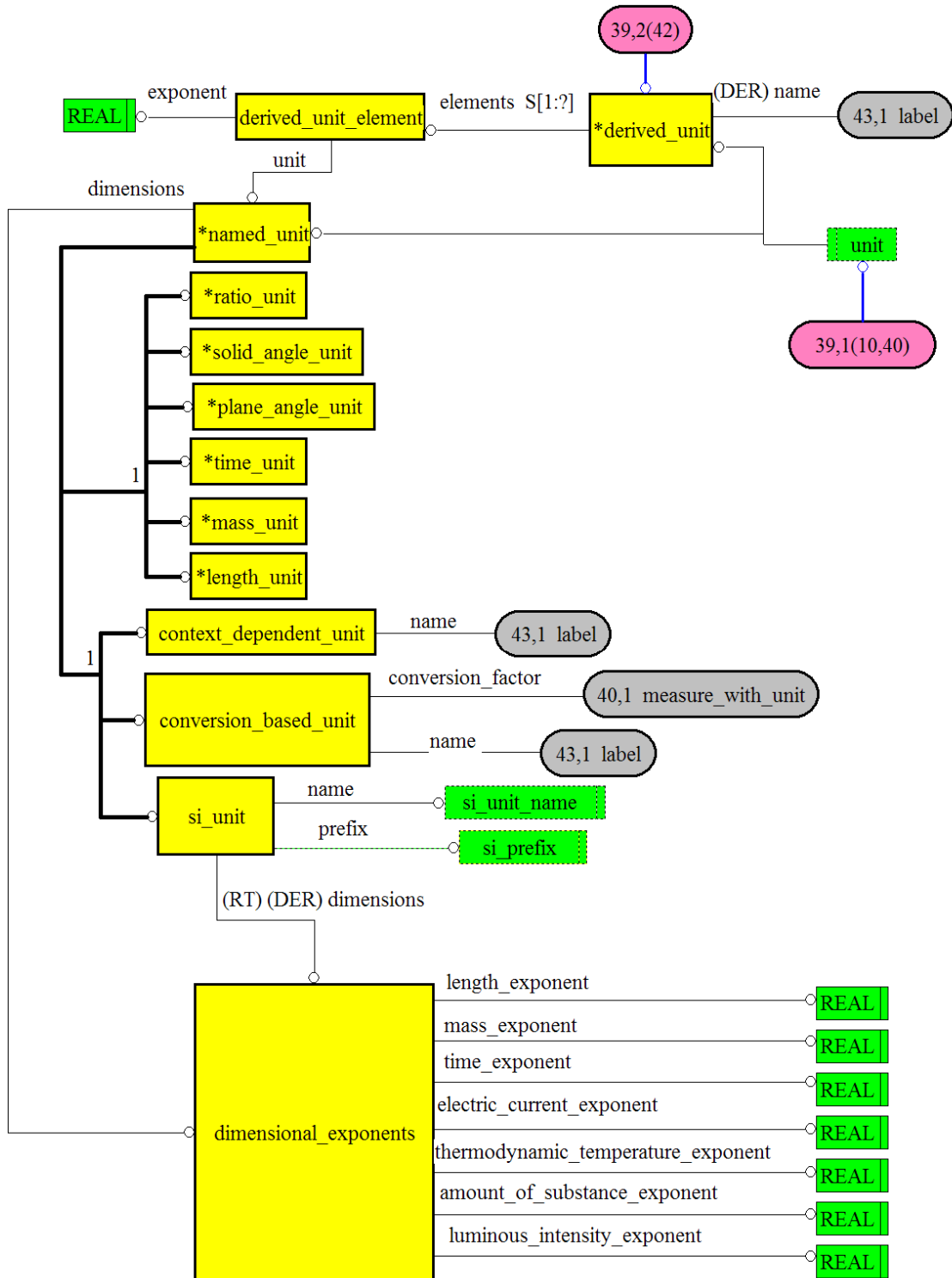


Figure H.39 — AIM EXPRESS-G diagram unit

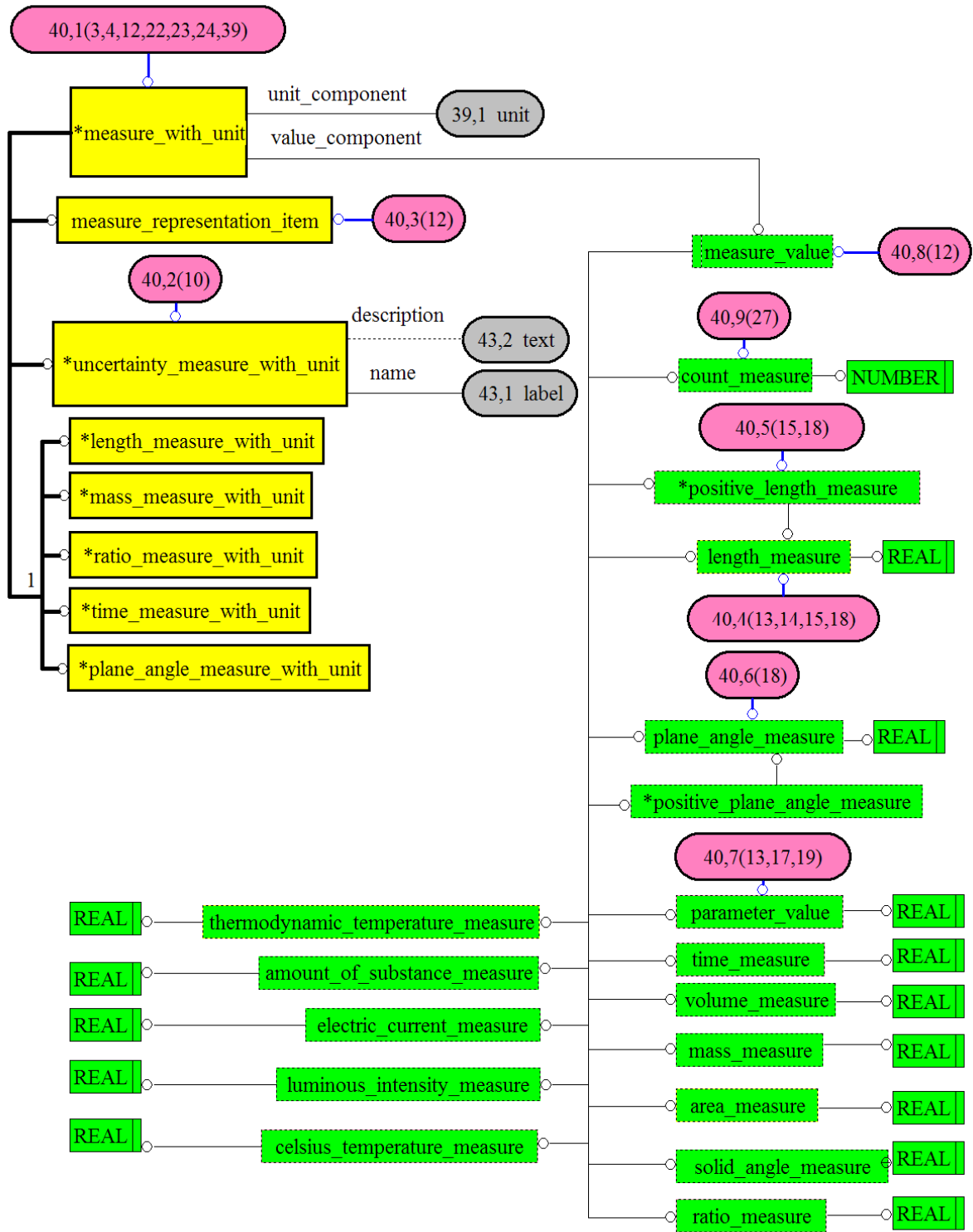
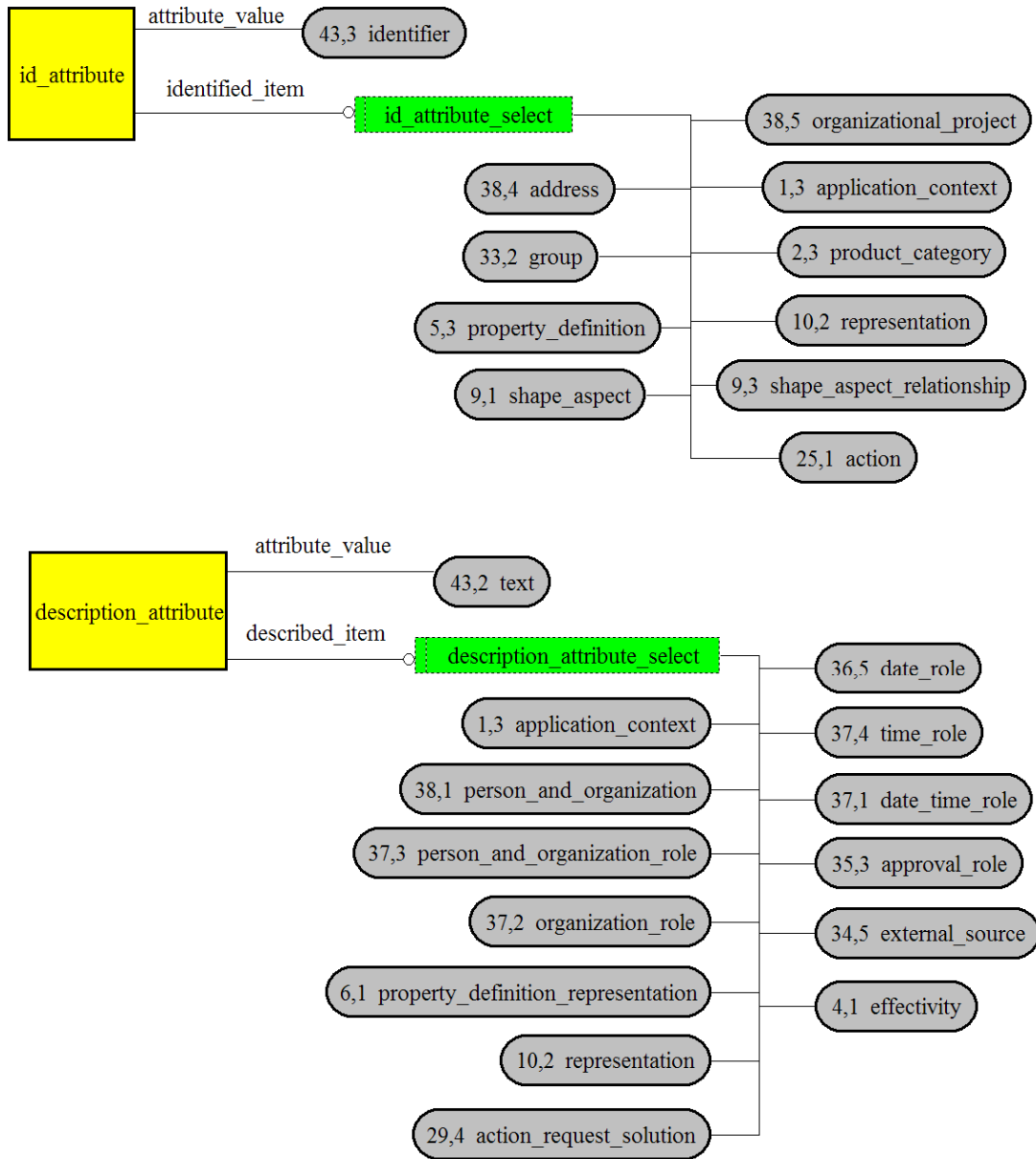
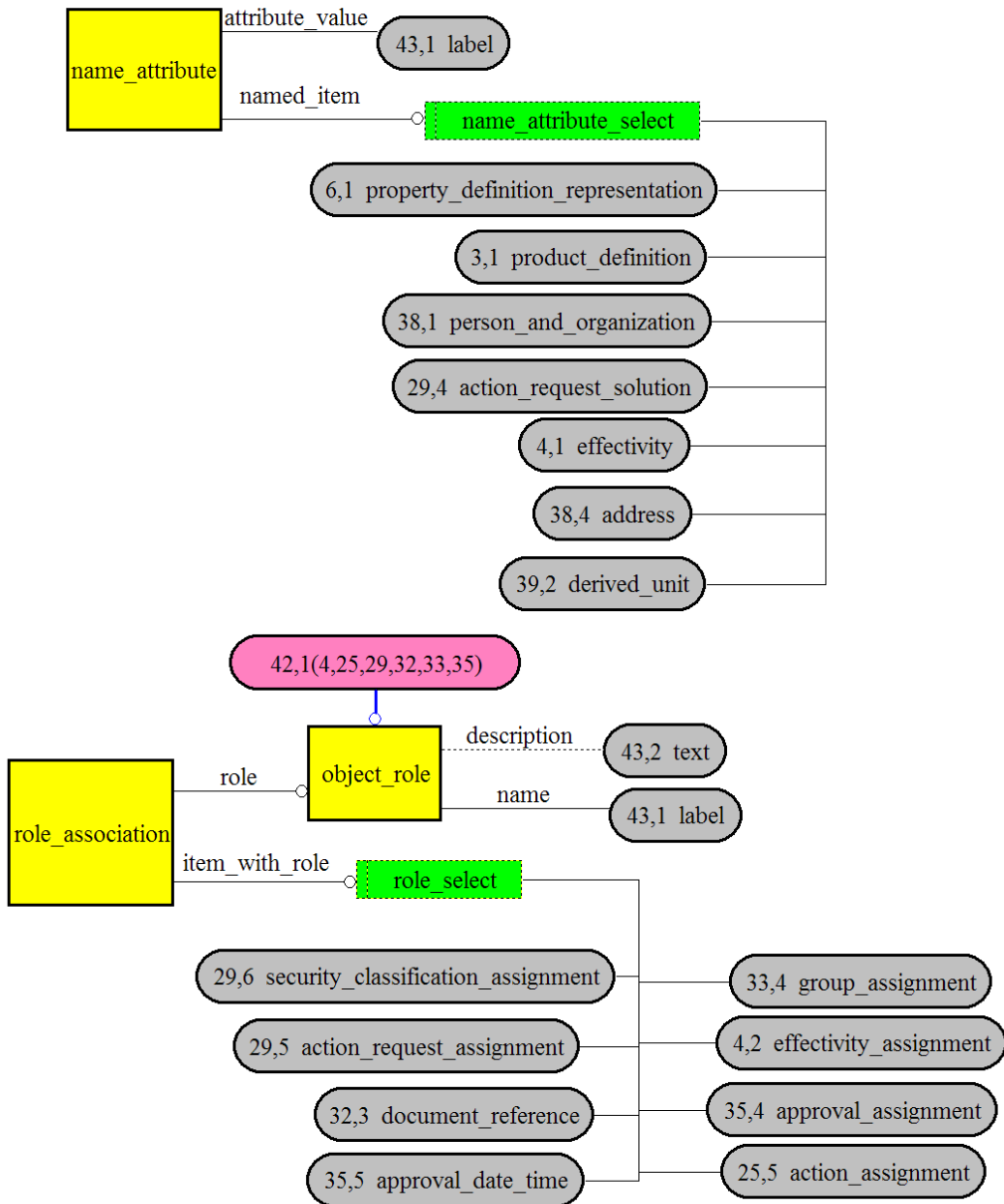


Figure H.40 — AIM EXPRESS-G diagram measure\_with\_unit

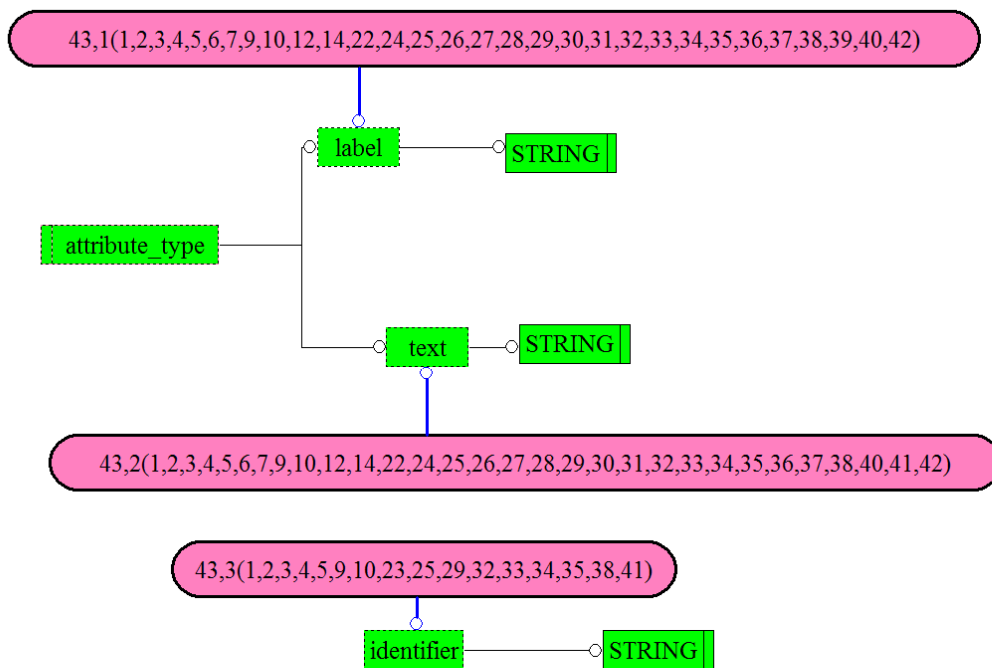


**Figure H.41 — AIM EXPRESS-G diagram id and description attribute**



**Figure H.42 — AIM EXPRESS-G diagram name and role attribute**





**Figure H.43 — AIM EXPRESS-G diagram text, label and identifier**

## **Annex I**

### **(informative)**

#### **Computer interpretable listings**

This annex provides a listing of the complete EXPRESS schema specified in annex A of this part of ISO 10303 without comments or other explanatory text. It also provides a listing of the EXPRESS entity names and corresponding short names as specified in annex B of this part of ISO 10303. The content of this annex is available in computer-interpretable form and can be found at the following URLs:

Short names: [http://www.tc184-sc4.org/Short\\_Names](http://www.tc184-sc4.org/Short_Names)

EXPRESS: <http://www.tc184-sc4.org/EXPRESS>

If there is difficulty accessing these sites contact ISO Central Secretariat or contact the ISO TC 184/SC 4 Secretariat directly at: [sc4sec@tc184-sc4.org](mailto:sc4sec@tc184-sc4.org).

NOTE The information provided in computer-interpretable form at the above URLs is informative. The information that is contained in the body of this part of ISO 10303 is normative.

## Annex J

### (informative)

## Application protocol usage guide

### J.1 Usage guidelines and test case definitions

The usage guide is documentation that describes the guidelines describing the usage scenarios, and/or example test cases. This documentation will be developed for the draft international standard of this part of ISO 10303.

### J.2 Purpose

This annex contains ISO 10303-21 listings showing how to create data instances that meet the requirements documented in the Mapping Tables for each of the objects in the ARM. Implementors can use these listings as an aid to understanding the representation of each ARM object, and as a set of example instances that can be copied by applications that want to create data sets for this part of ISO 10303.

### J.3 Syntax conventions

Each data set is described as a Part 21 exchange file containing a sequence of AIM instances with comments. The comments show the implementor what AIM instances shall be created to represent an ARM concept. The comments give lists of instance numbers used for a particular path. The instances and arrangement is formally defined by the mapping tables in 5.1.

The AIM data part contains instances that meet the requirements described by the mapping table. This listing shows the implementor how the AIM instances for each ARM concept shall be linked and what constant values shall be set for those instances. When a particular text value is required by the mapping table, it is underlined.

### J.4 Rationale for the selection of example parts

Geometry – This AP uses the same geometry as Part 240, Part 238 and Part 224. Geometry has been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM geometry is validated.

Dimensional and Geometric tolerances - This AP uses the same tolerances as Part 238, Part 240 and Part 224. Tolerances have been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM tolerance is validated.

Machining Features - This AP uses the same features as Part 240, Part 238 and Part 224. Features have been thoroughly validated both from an ARM and AIM perspective. Therefore The a test part will be used as an example that AIM features is validated.

Part 224 overlap - There are several UoFs that are in this AP that are common with Part 224: design\_exception, feature\_definition\_item, feature\_profile, library\_reference, manufacturing\_feature, manufacturing\_part\_properties, manufacturing\_process\_control\_documentation, measurement\_limitations, part\_model, requisitions, shape\_representation\_for\_casting\_and\_machining. Since these concepts have be thoroughly tested in Part 224 , the part will serve as an example to valide these concepts on the AIM schema.

Part 240 overlap - There are several UoFs that are in this part of ISO 10303 that are common with Part 240: manufacturing\_process\_requirement\_document, manufacturing\_process\_control\_documentation, part\_administration\_data, part\_model, process\_activities, process plan. Since these concepts have been thoroughly tested in Part 240, this part of ISO 10303 will concontrat on the object unique to this schema.

Casting features, die mould features, investment features – These features are unique to this Part, however they are developed with identical integration principals as machining features in Part 224. Therefoe the will be a test example for these features, the confidence of validation is extreamly high.

Casting prerequisites, conformance types, and simulation – These units of functionality are unique to this AP and there is no previous testing of these concepts by any other manufacturing application protocol. These UoFs will be the primary focus of testing for this Part.

## J.5 Cast part example

This test case was generated by hand population. This test does not validate the accuracy of the value, but the structure of the data and validation of entity rules and schema global rules. The intent is to validate that the Part 223 schema supports Part 223 data. This part was successfully tested against the schema in this document.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('AP224 File'),'2;1');
FILE_NAME('92824_800172_g.224','1997-08-01T11:07:18-05:00',('RPTS
Operator'),('RPTS'),'RPTS MP 6.0','PTC Pro/ENGINEER Version 18.0','RPTS
Operator');
FILE_SCHEMA(('CAST_PARTS_SCHEMA'));
ENDSEC;
DATA;
#5=DIMENSIONAL_EXPONENTS(1.0,0.0,0.0,0.0,0.0,0.0,0.0);
#6=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#8=DIMENSIONAL_EXPONENTS(0.0,0.0,1.0,0.0,0.0,0.0,0.0);
#10=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#12=RATIO_UNIT(#6);
#13=TIME_UNIT(#8);
#11=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(25.39999999999999),#10);
#15=(CONVERSION_BASED_UNIT('INCH',#11)LENGTH_UNIT()NAMED_UNIT(#5));
#19=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
```

```

#20=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.017453292519943),#19)
;
#24=(CONVERSION_BASED_UNIT('DEGREE',#20)NAMED_UNIT(#6)PLANE_ANGLE_UNIT());
#28=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#29=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.000001000000000),#15,'closure','maximum model space distance between geometric entities at asserted connectivities');
#34=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#29))GLOBAL_UNIT_ASSIGNED_CONTEXT((#15,#24,#28))REPRESENTATION_CONTEXT('ID1','3D'));
#35=CONTEXT_DEPENDENT_UNIT(#6,'COUNT');
#36=APPLICATION_CONTEXT('cast parts schema');
#37=APPLICATION_PROTOCOL_DEFINITION('Draft International Standard','cast parts schema',1996,#36);
#38=PRODUCT_CONTEXT('cast parts schema',#36,'Mechanical');
#39=PRODUCT('','',',',(#38));
#40=PRODUCT_DEFINITION_FORMATION('','',#39);
#41=PRODUCT_DEFINITION_CONTEXT('part definition',#36,'manufacturing planning');
#42=CAST_PART('','product design',#40,#41);
#43=PRODUCT_DEFINITION_SHAPE('product shape','shape for product',#42);
#44=CARTESIAN_POINT('',(-0.031000000000000,-0.183398473276088,0.0));
#45=CARTESIAN_POINT('',(-0.031000000000000,0.183398473276088,5.847965E-34));
#46=VERTEX_POINT('',#44);
#47=VERTEX_POINT('',#45);
#48=DIRECTION('',(0.0,1.0,0.0));
#49=VECTOR('',#48,0.366796946552176);
#50=LINE('',#44,#49);
#51=EDGE_CURVE('',#46,#47,#50,.T.);
#52=ORIENTED_EDGE('',*,*,#51,.T.);
#53=CARTESIAN_POINT('',(-0.186000000000000,0.0,0.0));
#54=VERTEX_POINT('',#53);
#55=CARTESIAN_POINT('',(0.0,0.0,0.0));
#56=DIRECTION('',(0.0,0.0,1.000000000000000));
#57=DIRECTION('',(-0.166666666666667,0.986013297183269,0.0));
#58=AXIS2_PLACEMENT_3D('',#55,#56,#57);
#59=CIRCLE('',#58,0.186000000000000);
#60=EDGE_CURVE('',#47,#54,#59,.T.);
#61=ORIENTED_EDGE('',*,*,#60,.T.);
#62=CARTESIAN_POINT('',(0.0,0.0,0.0));
#63=DIRECTION('',(0.0,0.0,1.0));
#64=DIRECTION('',(-1.0,0.0,0.0));
#65=AXIS2_PLACEMENT_3D('',#62,#63,#64);
#66=CIRCLE('',#65,0.186000000000000);
#67=EDGE_CURVE('',#54,#46,#66,.T.);
#68=ORIENTED_EDGE('',*,*,#67,.T.);
#69=EDGE_LOOP('',(#52,#61,#68));
#70=FACE_BOUND('',#69,.F.);
#71=CARTESIAN_POINT('',(0.0,0.0,0.0));
#72=DIRECTION('',(0.0,0.0,1.0));
#73=DIRECTION('',(1.0,0.0,0.0));
#74=AXIS2_PLACEMENT_3D('',#71,#72,#73);
#75=PLANE('',#74);

```

```

#76=ADVANCED_FACE('', (#70), #75, .F.);
#77=SHAPE_ASPECT('featured shape: face 1', '', #43, .T.);
#78=PROPERTY_DEFINITION('', '', #77);
#79=FACE_SHAPE_REPRESENTATION('', (#76), #34);
#80=SHAPE_DEFINITION_REPRESENTATION(#78, #79);
#81=CARTESIAN_POINT('', (0.0310000000000000, 0.183398473276088, 0.0));
#82=CARTESIAN_POINT('', (0.0310000000000000, -0.183398473276088, 0.0));
#83=VERTEX_POINT('', #81);
#84=VERTEX_POINT('', #82);
#85=DIRECTION('', (0.0, 1.0, 0.0));
#86=VECTOR('', #85, 0.366796946552176);
#87=LINE('', #82, #86);
#88=EDGE_CURVE('', #83, #84, #87, .F.);
#89=ORIENTED_EDGE('', *, *, #88, .T.);
#90=CARTESIAN_POINT('', (0.1860000000000000, 0.0, 0.0));
#91=VERTEX_POINT('', #90);
#92=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#93=DIRECTION('', (0.0, 0.0, 1.0000000000000000));
#94=DIRECTION('', (0.1666666666666667, -0.986013297183269, 0.0));
#95=AXIS2_PLACEMENT_3D('', #92, #93, #94);
#96=CIRCLE('', #95, 0.1860000000000000);
#97=EDGE_CURVE('', #84, #91, #96, .T.);
#98=ORIENTED_EDGE('', *, *, #97, .T.);
#99=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#100=DIRECTION('', (0.0, 0.0, 1.0));
#101=DIRECTION('', (1.0, 0.0, 0.0));
#102=AXIS2_PLACEMENT_3D('', #99, #100, #101);
#103=CIRCLE('', #102, 0.1860000000000000);
#104=EDGE_CURVE('', #91, #83, #103, .T.);
#105=ORIENTED_EDGE('', *, *, #104, .T.);
#106=EDGE_LOOP('', (#89, #98, #105));
#107=FACE_BOUND('', #106, .F.);
#108=ADVANCED_FACE('', (#107), #75, .F.);
#109=SHAPE_ASPECT('featured shape: face 2', '', #43, .T.);
#110=PROPERTY_DEFINITION('', '', #109);
#111=FACE_SHAPE_REPRESENTATION('', (#108), #34);
#112=SHAPE_DEFINITION_REPRESENTATION(#110, #111);
#113=CARTESIAN_POINT('', (0.0, 0.090166604983954, 4.3100000000000000));
#114=CARTESIAN_POINT('', (0.0, -0.090166604983954, 4.3100000000000000));
#115=VERTEX_POINT('', #113);
#116=VERTEX_POINT('', #114);
#117=CARTESIAN_POINT('', (0.0, 0.0, 4.3100000000000000));
#118=DIRECTION('', (0.0, 0.0, 1.0));
#119=DIRECTION('', (0.0, -1.0, 0.0));
#120=AXIS2_PLACEMENT_3D('', #117, #118, #119);
#121=CIRCLE('', #120, 0.090166604983954);
#122=EDGE_CURVE('', #115, #116, #121, .F.);
#123=ORIENTED_EDGE('', *, *, #122, .T.);
#124=CARTESIAN_POINT('', (0.0, 0.0, 4.3100000000000000));
#125=DIRECTION('', (0.0, 0.0, 1.0));
#126=DIRECTION('', (0.0, 1.0, 0.0));
#127=AXIS2_PLACEMENT_3D('', #124, #125, #126);
#128=CIRCLE('', #127, 0.090166604983954);

```

```

#129=EDGE_CURVE('',#116,#115,#128,.F.);
#130=ORIENTED_EDGE('',*,*,#129,.T.);
#131=EDGE_LOOP('',(#123,#130));
#132=FACE_BOUND('',#131,.F.);
#133=CARTESIAN_POINT('',(0.0,0.0,4.310000000000000));
#134=DIRECTION('',(0.0,0.0,1.0));
#135=DIRECTION('',(1.0,0.0,0.0));
#136=AXIS2_PLACEMENT_3D('',#133,#134,#135);
#137=PLANE('',#136);
#138=ADVANCED_FACE('',(#132),#137,.T.);
#139=SHAPE_ASPECT('featured shape: face 3','',#43,.T.);
#140=PROPERTY_DEFINITION('',',',#139);
#141=FACE_SHAPE_REPRESENTATION('',(#138),#34);
#142=SHAPE_DEFINITION_REPRESENTATION(#140,#141);
#143=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.580000000000000
0));
#144=CARTESIAN_POINT('',(0.186000000000000,0.0,2.642500000000000));
#145=VERTEX_POINT('',#143);
#146=VERTEX_POINT('',#144);
#147=CARTESIAN_POINT('',(0.186000000000000,0.0,2.642500000000000));
#148=CARTESIAN_POINT('',(0.186000000000000,0.005471621857183,2.642500000000000
0));
#149=CARTESIAN_POINT('',(0.185536257371889,0.016107401839008,2.64114590848819
1));
#150=CARTESIAN_POINT('',(0.183444961303938,0.031895399078690,2.63461599318274
4));
#151=CARTESIAN_POINT('',(0.180647351963460,0.044822005907028,2.62456383972068
9));
#152=CARTESIAN_POINT('',(0.177772939484603,0.054932562783282,2.61133932574037
7));
#153=CARTESIAN_POINT('',(0.175676556620784,0.061146675813468,2.59594226183308
3));
#154=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.58542285607518
0));
#155=CARTESIAN_POINT('',(0.175184902317523,0.062500000000000,2.580000000000000
0));
#156=B_SPLINE_CURVE_WITH_KNOTS('',3,(#147,#148,#149,#150,#151,#152,#153,#154,
#155),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#157=EDGE_CURVE('',#145,#146,#156,.F.);
#158=ORIENTED_EDGE('',*,*,#157,.T.);
#159=CARTESIAN_POINT('',(0.186000000000000,0.0,3.800000000000000));
#160=VERTEX_POINT('',#159);
#161=DIRECTION('',(0.0,0.0,1.0));
#162=VECTOR('',#161,1.157500000000000);
#163=LINE('',#144,#162);
#164=EDGE_CURVE('',#146,#160,#163,.T.);
#165=ORIENTED_EDGE('',*,*,#164,.T.);
#166=CARTESIAN_POINT('',(1.138922E-17,0.186000000000000,3.800000000000000));
#167=VERTEX_POINT('',#166);
#168=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#169=DIRECTION('',(0.0,0.0,1.0));

```

```

#170=DIRECTION('', (1.0,0.0,0.0));
#171=AXIS2_PLACEMENT_3D('', #168, #169, #170);
#172=CIRCLE('', #171, 0.1860000000000000);
#173=EDGE_CURVE('', #160, #167, #172, .T.);
#174=ORIENTED_EDGE('', *, *, #173, .T.);
#175=CARTESIAN_POINT('', (-0.1860000000000000, -5.982216E-
17, 3.8000000000000000));
#176=VERTEX_POINT('', #175);
#177=CARTESIAN_POINT('', (0.0, 0.0, 3.7999999999999999));
#178=DIRECTION('', (0.0, 0.0, 1.0));
#179=DIRECTION('', (6.123234E-17, 1.0, 0.0));
#180=AXIS2_PLACEMENT_3D('', #177, #178, #179);
#181=CIRCLE('', #180, 0.1860000000000000);
#182=EDGE_CURVE('', #167, #176, #181, .T.);
#183=ORIENTED_EDGE('', *, *, #182, .T.);
#184=CARTESIAN_POINT('', (-0.1860000000000000, 2.277843E-17, 2.6425000000000000));
#185=VERTEX_POINT('', #184);
#186=DIRECTION('', (0.0, 0.0, 1.0));
#187=VECTOR('', #186, 1.1575000000000000);
#188=LINE('', #184, #187);
#189=EDGE_CURVE('', #176, #185, #188, .F.);
#190=ORIENTED_EDGE('', *, *, #189, .T.);
#191=CARTESIAN_POINT('', (-
0.175184902317523, 0.0625000000000000, 2.5800000000000001));
#192=VERTEX_POINT('', #191);
#193=CARTESIAN_POINT('', (-
0.175184902317523, 0.0625000000000000, 2.5800000000000001));
#194=CARTESIAN_POINT('', (-
0.175184902317523, 0.0625000000000000, 2.585426709005403));
#195=CARTESIAN_POINT('', (-
0.175677115146707, 0.061145342990551, 2.595951978028653));
#196=CARTESIAN_POINT('', (-
0.177776266176458, 0.054921858431442, 2.611358030685044));
#197=CARTESIAN_POINT('', (-
0.180650079241388, 0.044811772210325, 2.624574571398843));
#198=CARTESIAN_POINT('', (-
0.183449265169815, 0.031871520572190, 2.634630402017820));
#199=CARTESIAN_POINT('', (-
0.185537370272068, 0.016090248458853, 2.641148978803535));
#200=CARTESIAN_POINT('', (-
0.1860000000000000, 0.005464543947207, 2.6425000000000000));
#201=CARTESIAN_POINT('', (-0.1860000000000000, 2.277843E-17, 2.6425000000000000));
#202=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#193, #194, #195, #196, #197, #198, #199, #200,
#201), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 4), (0.0, 0.1666666666666667, 0.33333
3333333333, 0.5000000000000000, 0.6666666666666667, 0.8333333333333333, 1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#203=EDGE_CURVE('', #185, #192, #202, .F.);
#204=ORIENTED_EDGE('', *, *, #203, .T.);
#205=CARTESIAN_POINT('', (-0.1860000000000000, 2.277843E-17, 2.5175000000000000));
#206=VERTEX_POINT('', #205);
#207=CARTESIAN_POINT('', (-0.1860000000000000, 2.277843E-17, 2.5175000000000000));
#208=CARTESIAN_POINT('', (-
0.1860000000000000, 0.005458851337490, 2.5175000000000000));

```



```

#209=CARTESIAN_POINT('', (-
0.185536093618712, 0.016073651023862, 2.518856289865521));
#210=CARTESIAN_POINT('', (-
0.183470250159010, 0.031747433690688, 2.525300488417541));
#211=CARTESIAN_POINT('', (-
0.180661972444195, 0.044771773357708, 2.535375964999431));
#212=CARTESIAN_POINT('', (-
0.177781605611539, 0.054904708977402, 2.548606047863121));
#213=CARTESIAN_POINT('', (-
0.175674738176842, 0.061152720479089, 2.564064031685822));
#214=CARTESIAN_POINT('', (-
0.175184902317523, 0.062500000000000, 2.574581881011206));
#215=CARTESIAN_POINT('', (-
0.175184902317523, 0.062500000000000, 2.580000000000001));
#216=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#207, #208, #209, #210, #211, #212, #213, #214,
#215), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 4), (0.0, 0.166666666666667, 0.33333
3333333333, 0.500000000000000, 0.666666666666667, 0.833333333333333, 1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#217=EDGE_CURVE('', #192, #206, #216, .F.);
#218=ORIENTED_EDGE('', *, *, #217, .T.);
#219=CARTESIAN_POINT('', (-0.186000000000000, 2.277843E-17, 0.344000000000001));
#220=VERTEX_POINT('', #219);
#221=DIRECTION('', (0.0, 0.0, 1.0));
#222=VECTOR('', #221, 2.173499999999999);
#223=LINE('', #219, #222);
#224=EDGE_CURVE('', #206, #220, #223, .F.);
#225=ORIENTED_EDGE('', *, *, #224, .T.);
#226=CARTESIAN_POINT('', (-
0.160499221181911, 0.094000000000000, 0.250000000000000));
#227=VERTEX_POINT('', #226);
#228=CARTESIAN_POINT('', (-
0.160499221181911, 0.094000000000000, 0.250000000000000));
#229=CARTESIAN_POINT('', (-
0.160499221181911, 0.094000000000000, 0.256766446956972));
#230=CARTESIAN_POINT('', (-
0.161327432070906, 0.092615564162586, 0.270119577772784));
#231=CARTESIAN_POINT('', (-
0.165192932924788, 0.085677048896482, 0.290595468399896));
#232=CARTESIAN_POINT('', (-
0.170610275358273, 0.074562472136122, 0.308562918733388));
#233=CARTESIAN_POINT('', (-
0.176618325764372, 0.059201828025679, 0.324055274565888));
#234=CARTESIAN_POINT('', (-
0.181743103977182, 0.041251906320261, 0.335390883863212));
#235=CARTESIAN_POINT('', (-
0.185283487333834, 0.020349156037856, 0.342600715969244));
#236=CARTESIAN_POINT('', (-
0.186000000000000, 0.006811219284266, 0.344000000000000));
#237=CARTESIAN_POINT('', (-0.186000000000000, 2.277843E-17, 0.344000000000000));
#238=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#228, #229, #230, #231, #232, #233, #234, #235,
#236, #237), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 4), (0.0, 0.142857142857143,
0.285714285714286, 0.428571428571429, 0.571428571428571, 0.714285714285714, 0.857
142857142857, 1.0), .PIECEWISE_BEZIER_KNOTS.);

```

```

#239=EDGE_CURVE('',#220,#227,#238,.F.);
#240=ORIENTED_EDGE('',*,*,#239,.T.);
#241=CARTESIAN_POINT('',(-0.1860000000000000,2.277843E-17,0.1560000000000001));
#242=VERTEX_POINT('',#241);
#243=CARTESIAN_POINT('',(-0.1860000000000000,2.277843E-17,0.1560000000000001));
#244=CARTESIAN_POINT('',(-
0.1860000000000000,0.006800366988429,0.1560000000000001));
#245=CARTESIAN_POINT('',(-
0.185284237508996,0.020320470184615,0.157398064439927));
#246=CARTESIAN_POINT('',(-
0.181763787252156,0.041157668374866,0.164566419180857));
#247=CARTESIAN_POINT('',(-
0.176635182649387,0.059160281094706,0.175901786861670));
#248=CARTESIAN_POINT('',(-
0.170601634807429,0.074584180536545,0.191460403648173));
#249=CARTESIAN_POINT('',(-
0.165181827617091,0.085698349461980,0.209438767375007));
#250=CARTESIAN_POINT('',(-
0.161310523171716,0.092644858063427,0.230008460101478));
#251=CARTESIAN_POINT('',(-
0.160499221181911,0.0940000000000000,0.243288192214063));
#252=CARTESIAN_POINT('',(-
0.160499221181911,0.0940000000000000,0.2500000000000000));
#253=B_SPLINE_CURVE_WITH_KNOTS('',3,(#243,#244,#245,#246,#247,#248,#249,#250,
#251,#252),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#254=EDGE_CURVE('',#227,#242,#253,.F.);
#255=ORIENTED_EDGE('',*,*,#254,.T.);
#256=DIRECTION('',(0.0,0.0,1.0));
#257=VECTOR('',#256,0.1560000000000000);
#258=LINE('',#53,#257);
#259=EDGE_CURVE('',#242,#54,#258,.F.);
#260=ORIENTED_EDGE('',*,*,#259,.T.);
#261=ORIENTED_EDGE('',*,*,#60,.F.);
#262=CARTESIAN_POINT('',(-
0.0310000000000000,0.183398473276088,0.6175000000000000));
#263=VERTEX_POINT('',#262);
#264=DIRECTION('',(-6.742245E-17,0.0,1.0));
#265=VECTOR('',#264,0.6175000000000000);
#266=LINE('',#45,#265);
#267=EDGE_CURVE('',#47,#263,#266,.T.);
#268=ORIENTED_EDGE('',*,*,#267,.T.);
#269=CARTESIAN_POINT('',(-
0.0185000000000000,0.185077686391418,0.6300000000000000));
#270=VERTEX_POINT('',#269);
#271=CARTESIAN_POINT('',(-
0.0185000000000000,0.185077686391418,0.6300000000000000));
#272=CARTESIAN_POINT('',(-
0.020222806907318,0.184905478015536,0.6300000000000000));
#273=CARTESIAN_POINT('',(-
0.023380441464034,0.184543239000241,0.629340957979689));

```

```

#274=CARTESIAN_POINT('', (-
0.027542467419886, 0.183961116646317, 0.626594726403956));
#275=CARTESIAN_POINT('', (-
0.030285994332659, 0.183520653386331, 0.622509121211844));
#276=CARTESIAN_POINT('', (-
0.031000000000000, 0.183398473276088, 0.619276684284376));
#277=CARTESIAN_POINT('', (-
0.031000000000000, 0.183398473276088, 0.617500000000000));
#278=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#271, #272, #273, #274, #275, #276, #277), .HYP
ERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 4), (0.0, 0.250000000000000, 0.500000000000000, 0.7
500000000000000, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#279=EDGE_CURVE('', #263, #270, #278, .F.);
#280=ORIENTED_EDGE('', *, *, #279, .T.);
#281=CARTESIAN_POINT('', (0.018500000000000, 0.185077686391418, 0.630000000000000
0));
#282=VERTEX_POINT('', #281);
#283=CARTESIAN_POINT('', (0.0, 0.0, 0.630000000000000));
#284=DIRECTION('', (0.0, 0.0, -1.0));
#285=DIRECTION('', (-0.099462365591398, 0.995041324685042, 0.0));
#286=AXIS2_PLACEMENT_3D('', #283, #284, #285);
#287=CIRCLE('', #286, 0.186000000000000);
#288=EDGE_CURVE('', #270, #282, #287, .T.);
#289=ORIENTED_EDGE('', *, *, #288, .T.);
#290=CARTESIAN_POINT('', (0.031000000000000, 0.183398473276088, 0.617500000000000
0));
#291=VERTEX_POINT('', #290);
#292=CARTESIAN_POINT('', (0.031000000000000, 0.183398473276088, 0.617500000000000
0));
#293=CARTESIAN_POINT('', (0.031000000000000, 0.183398473276088, 0.61920375881085
9));
#294=CARTESIAN_POINT('', (0.030354567108582, 0.183509179238234, 0.62233725817082
5));
#295=CARTESIAN_POINT('', (0.027638547132190, 0.183946283585950, 0.62650045564978
7));
#296=CARTESIAN_POINT('', (0.023535786372608, 0.184523602834953, 0.62927917584640
3));
#297=CARTESIAN_POINT('', (0.020288448038623, 0.184898916657693, 0.630000000000000
0));
#298=CARTESIAN_POINT('', (0.018500000000000, 0.185077686391418, 0.630000000000000
0));
#299=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#292, #293, #294, #295, #296, #297, #298), .HYP
ERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 4), (0.0, 0.250000000000000, 0.500000000000000, 0.7
500000000000000, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#300=EDGE_CURVE('', #282, #291, #299, .F.);
#301=ORIENTED_EDGE('', *, *, #300, .T.);
#302=DIRECTION('', (0.0, 0.0, 1.0));
#303=VECTOR('', #302, 0.617500000000000);
#304=LINE('', #81, #303);
#305=EDGE_CURVE('', #291, #83, #304, .F.);
#306=ORIENTED_EDGE('', *, *, #305, .T.);
#307=ORIENTED_EDGE('', *, *, #104, .F.);
#308=CARTESIAN_POINT('', (0.186000000000000, 0.0, 0.156000000000000));
#309=VERTEX_POINT('', #308);

```

ISO 10303-223:2008 (E)

```
#310=DIRECTION('', (0.0,0.0,1.0));
#311=VECTOR('', #310,0.1560000000000000);
#312=LINE('', #90, #311);
#313=EDGE_CURVE('', #91, #309, #312, .T.);
#314=ORIENTED_EDGE('', *, *, #313, .T.);
#315=CARTESIAN_POINT('', (0.160499221181911,0.0940000000000000,0.2500000000000000
0));
#316=VERTEX_POINT('', #315);
#317=CARTESIAN_POINT('', (0.160499221181911,0.0940000000000000,0.2500000000000000
0));
#318=CARTESIAN_POINT('', (0.160499221181911,0.0940000000000000,0.24329136870794
2));
#319=CARTESIAN_POINT('', (0.161309755247970,0.092646107527992,0.23001745040141
5));
#320=CARTESIAN_POINT('', (0.165177934867762,0.085705567809687,0.20945550416081
0));
#321=CARTESIAN_POINT('', (0.170594327073985,0.074600077742975,0.19148197206178
3));
#322=CARTESIAN_POINT('', (0.176625064117123,0.059189393219094,0.17592576770399
5));
#323=CARTESIAN_POINT('', (0.181755664328797,0.041193271352137,0.16458337327453
6));
#324=CARTESIAN_POINT('', (0.185281410572974,0.020351551813128,0.15740359268013
0));
#325=CARTESIAN_POINT('', (0.1860000000000000,0.006812818398979,0.1560000000000000
0));
#326=CARTESIAN_POINT('', (0.1860000000000000,0.0,0.1560000000000000));
#327=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#317, #318, #319, #320, #321, #322, #323, #324,
#325, #326), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 1, 4), (0.0, 0.142857142857143,
0.285714285714286, 0.428571428571429, 0.571428571428571, 0.714285714285714, 0.857
142857142857, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#328=EDGE_CURVE('', #309, #316, #327, .F.);
#329=ORIENTED_EDGE('', *, *, #328, .T.);
#330=CARTESIAN_POINT('', (0.1860000000000000,0.0,0.3440000000000000));
#331=VERTEX_POINT('', #330);
#332=CARTESIAN_POINT('', (0.1860000000000000,0.0,0.3440000000000000));
#333=CARTESIAN_POINT('', (0.1860000000000000,0.006823032072908,0.3440000000000000
0));
#334=CARTESIAN_POINT('', (0.185280833739913,0.020378752353648,0.34259554757765
5));
#335=CARTESIAN_POINT('', (0.181735072423795,0.041287079595952,0.33537407462788
6));
#336=CARTESIAN_POINT('', (0.176608347816713,0.059230733088108,0.32403177810720
3));
#337=CARTESIAN_POINT('', (0.170601713482259,0.074581303037190,0.30853777493501
9));
#338=CARTESIAN_POINT('', (0.165188657251781,0.085684909394755,0.29057713245411
6));
#339=CARTESIAN_POINT('', (0.161326264069271,0.092617528844491,0.27010754428205
9));
#340=CARTESIAN_POINT('', (0.160499221181911,0.0940000000000000,0.25676207178187
0));
```

```

#341=CARTESIAN_POINT('', (0.160499221181911, 0.0940000000000000, 0.2500000000000000
0));
#342=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#332, #333, #334, #335, #336, #337, #338, #339,
#340, #341), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 1, 4), (0.0, 0.142857142857143,
0.285714285714286, 0.428571428571429, 0.571428571428571, 0.714285714285714, 0.857
142857142857, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#343=EDGE_CURVE('', #316, #331, #342, .F.);
#344=ORIENTED_EDGE('', *, *, #343, .T.);
#345=CARTESIAN_POINT('', (0.1860000000000000, 0.0, 2.5175000000000000));
#346=VERTEX_POINT('', #345);
#347=DIRECTION('', (0.0, 0.0, 1.0));
#348=VECTOR('', #347, 2.1735000000000000);
#349=LINE('', #330, #348);
#350=EDGE_CURVE('', #331, #346, #349, .T.);
#351=ORIENTED_EDGE('', *, *, #350, .T.);
#352=CARTESIAN_POINT('', (0.175184902317523, 0.0625000000000000, 2.580000000000000
0));
#353=CARTESIAN_POINT('', (0.175184902317523, 0.0625000000000000, 2.57458465168521
7));
#354=CARTESIAN_POINT('', (0.175674414168389, 0.061153355224084, 2.56407098903970
9));
#355=CARTESIAN_POINT('', (0.177778707786166, 0.054914219798778, 2.54862220454415
8));
#356=CARTESIAN_POINT('', (0.180660130799249, 0.044778201243767, 2.53538329137297
4));
#357=CARTESIAN_POINT('', (0.183465542301875, 0.031773272894993, 2.52531636727892
4));
#358=CARTESIAN_POINT('', (0.185534991382704, 0.016091351499283, 2.51885926065962
5));
#359=CARTESIAN_POINT('', (0.1860000000000000, 0.005466293856892, 2.517500000000000
0));
#360=CARTESIAN_POINT('', (0.1860000000000000, 0.0, 2.5175000000000000));
#361=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#352, #353, #354, #355, #356, #357, #358, #359,
#360), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 1, 4), (0.0, 0.166666666666667, 0.33333
3333333333, 0.5000000000000000, 0.666666666666667, 0.833333333333333, 1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#362=EDGE_CURVE('', #346, #145, #361, .F.);
#363=ORIENTED_EDGE('', *, *, #362, .T.);
#364=EDGE_LOOP('', (#158, #165, #174, #183, #190, #204, #218, #225, #240, #255, #260, #26
1, #268, #280, #289, #301, #306, #307, #314, #329, #344, #351, #363));
#365=FACE_BOUND('', #364, .F.);
#366=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#367=DIRECTION('', (0.0, 0.0, 1.0));
#368=DIRECTION('', (1.0, 0.0, 0.0));
#369=AXIS2_PLACEMENT_3D('', #366, #367, #368);
#370=CYLINDRICAL_SURFACE('', #369, 0.1860000000000000);
#371=ADVANCED_FACE('', (#365), #370, .T.);
#372=SHAPE_ASPECT('featured shape: face 4', '', #43, .T.);
#373=PROPERTY_DEFINITION('', '', #372);
#374=FACE_SHAPE_REPRESENTATION('', (#371), #34);
#375=SHAPE_DEFINITION_REPRESENTATION(#373, #374);
#376=CARTESIAN_POINT('', (-0.175184902317523, -
0.0625000000000000, 2.5800000000000000));

```

```

#377=VERTEX_POINT('',#376);
#378=CARTESIAN_POINT('',(-0.1860000000000000,2.277843E-17,2.642500000000000));
#379=CARTESIAN_POINT('',(-0.1860000000000000,-
0.005466293856892,2.642500000000000));
#380=CARTESIAN_POINT('',(-0.185534991382705,-
0.016091351499283,2.641140739340373));
#381=CARTESIAN_POINT('',(-0.183465542301875,-
0.031773272894993,2.634683632721078));
#382=CARTESIAN_POINT('',(-0.180660130799249,-
0.044778201243767,2.624616708627026));
#383=CARTESIAN_POINT('',(-0.177778707786166,-
0.054914219798778,2.611377795455842));
#384=CARTESIAN_POINT('',(-0.175674414168389,-
0.061153355224084,2.595929010960289));
#385=CARTESIAN_POINT('',(-0.175184902317523,-
0.0625000000000000,2.585415348314783));
#386=CARTESIAN_POINT('',(-0.175184902317523,-
0.0625000000000000,2.580000000000000));
#387=B_SPLINE_CURVE_WITH_KNOTS('',3,(#378,#379,#380,#381,#382,#383,#384,#385,
#386),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,4),(0.0,0.1666666666666667,0.33333
3333333333,0.5000000000000000,0.6666666666666667,0.8333333333333333,1.0),.PIECEW
ISE_BEZIER_KNOTS.);
#388=EDGE_CURVE('',#377,#185,#387,.F.);
#389=ORIENTED_EDGE('',*,*,#388,.T.);
#390=ORIENTED_EDGE('',*,*,#189,.F.);
#391=CARTESIAN_POINT('',(-3.416765E-17,-
0.1860000000000000,3.800000000000000));
#392=VERTEX_POINT('',#391);
#393=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#394=DIRECTION('',(0.0,0.0,1.0));
#395=DIRECTION('',(-1.0,1.224647E-16,0.0));
#396=AXIS2_PLACEMENT_3D('',#393,#394,#395);
#397=CIRCLE('',#396,0.1860000000000000);
#398=EDGE_CURVE('',#176,#392,#397,.T.);
#399=ORIENTED_EDGE('',*,*,#398,.T.);
#400=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#401=DIRECTION('',(0.0,0.0,1.0));
#402=DIRECTION('',(-1.836970E-16,-1.0,0.0));
#403=AXIS2_PLACEMENT_3D('',#400,#401,#402);
#404=CIRCLE('',#403,0.1860000000000000);
#405=EDGE_CURVE('',#392,#160,#404,.T.);
#406=ORIENTED_EDGE('',*,*,#405,.T.);
#407=ORIENTED_EDGE('',*,*,#164,.F.);
#408=CARTESIAN_POINT('',(0.175184902317523,-
0.0625000000000000,2.580000000000000));
#409=VERTEX_POINT('',#408);
#410=CARTESIAN_POINT('',(0.175184902317523,-
0.0625000000000000,2.580000000000000));
#411=CARTESIAN_POINT('',(0.175184902317523,-
0.0625000000000000,2.585418118988794));
#412=CARTESIAN_POINT('',(0.175674738176842,-
0.061152720479089,2.595935968314176));

```

```

#413=CARTESIAN_POINT('', (0.177781605611539,-
0.054904708977402,2.611393952136881));
#414=CARTESIAN_POINT('', (0.180661972444195,-
0.044771773357707,2.624624035000569));
#415=CARTESIAN_POINT('', (0.183470250159010,-
0.031747433690688,2.634699511582460));
#416=CARTESIAN_POINT('', (0.185536093618712,-
0.016073651023862,2.641143710134478));
#417=CARTESIAN_POINT('', (0.186000000000000,-
0.005458851337490,2.642500000000000));
#418=CARTESIAN_POINT('', (0.186000000000000,-4.555686E-17,2.642500000000000));
#419=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#410,#411,#412,#413,#414,#415,#416,#417,
#418), .HYPERBOLIC_ARC., .F., .F., (4,1,1,1,1,1,4), (0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#420=EDGE_CURVE('', #146,#409,#419,.F.);
#421=ORIENTED_EDGE('', *,*, #420,.T.);
#422=CARTESIAN_POINT('', (0.186000000000000,-4.555686E-17,2.517500000000000));
#423=CARTESIAN_POINT('', (0.186000000000000,-
0.005464543947207,2.517500000000000));
#424=CARTESIAN_POINT('', (0.185537370272068,-
0.016090248458853,2.518851021196464));
#425=CARTESIAN_POINT('', (0.183449265169815,-
0.031871520572190,2.525369597982180));
#426=CARTESIAN_POINT('', (0.180650079241389,-
0.044811772210324,2.535425428601157));
#427=CARTESIAN_POINT('', (0.177776266176458,-
0.054921858431443,2.548641969314958));
#428=CARTESIAN_POINT('', (0.175677115146707,-
0.061145342990551,2.564048021971347));
#429=CARTESIAN_POINT('', (0.175184902317523,-
0.062500000000000,2.574573290994597));
#430=CARTESIAN_POINT('', (0.175184902317523,-
0.062500000000000,2.580000000000000));
#431=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#422,#423,#424,#425,#426,#427,#428,#429,
#430), .HYPERBOLIC_ARC., .F., .F., (4,1,1,1,1,1,4), (0.0,0.166666666666667,0.33333
3333333333,0.500000000000000,0.666666666666667,0.833333333333333,1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#432=EDGE_CURVE('', #409,#346,#431,.F.);
#433=ORIENTED_EDGE('', *,*, #432,.T.);
#434=ORIENTED_EDGE('', *,*, #350,.F.);
#435=CARTESIAN_POINT('', (0.160499221181911,-
0.094000000000000,0.250000000000000));
#436=VERTEX_POINT('', #435);
#437=CARTESIAN_POINT('', (0.160499221181911,-
0.094000000000000,0.250000000000000));
#438=CARTESIAN_POINT('', (0.160499221181911,-
0.094000000000000,0.256711807785937));
#439=CARTESIAN_POINT('', (0.161310523171716,-
0.0926444858063427,0.269991539898522));
#440=CARTESIAN_POINT('', (0.165181827617091,-
0.085698349461980,0.290561232624993));

```

```

#441=CARTESIAN_POINT('', (0.170601634807429,-
0.074584180536545,0.308539596351827));
#442=CARTESIAN_POINT('', (0.176635182649387,-
0.059160281094706,0.324098213138330));
#443=CARTESIAN_POINT('', (0.181763787252156,-
0.041157668374865,0.335433580819143));
#444=CARTESIAN_POINT('', (0.185284237508996,-
0.020320470184614,0.342601935560072));
#445=CARTESIAN_POINT('', (0.186000000000000,-
0.006800366988429,0.344000000000000));
#446=CARTESIAN_POINT('', (0.186000000000000,-4.555686E-17,0.344000000000000));
#447=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#437,#438,#439,#440,#441,#442,#443,#444,
#445,#446), .HYPERBOLIC_ARC., .F., .F., (4,1,1,1,1,1,1,4), (0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0), .PIECEWISE_BEZIER_KNOTS.);
#448=EDGE_CURVE('', #331,#436,#447,.F.);
#449=ORIENTED_EDGE('', *,*, #448,.T.);
#450=CARTESIAN_POINT('', (0.186000000000000,-4.555686E-17,0.156000000000000));
#451=CARTESIAN_POINT('', (0.186000000000000,-
0.006811219284266,0.156000000000000));
#452=CARTESIAN_POINT('', (0.185283487333834,-
0.020349156037856,0.157399284030756));
#453=CARTESIAN_POINT('', (0.181743103977182,-
0.041251906320261,0.164609116136788));
#454=CARTESIAN_POINT('', (0.176618325764372,-
0.059201828025679,0.175944725434112));
#455=CARTESIAN_POINT('', (0.170610275358273,-
0.074562472136122,0.191437081266612));
#456=CARTESIAN_POINT('', (0.165192932924788,-
0.085677048896483,0.209404531600104));
#457=CARTESIAN_POINT('', (0.161327432070906,-
0.092615564162586,0.229880422227216));
#458=CARTESIAN_POINT('', (0.160499221181911,-
0.094000000000000,0.243233553043028));
#459=CARTESIAN_POINT('', (0.160499221181911,-
0.094000000000000,0.250000000000000));
#460=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#450,#451,#452,#453,#454,#455,#456,#457,
#458,#459), .HYPERBOLIC_ARC., .F., .F., (4,1,1,1,1,1,1,4), (0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0), .PIECEWISE_BEZIER_KNOTS.);
#461=EDGE_CURVE('', #436,#309,#460,.F.);
#462=ORIENTED_EDGE('', *,*, #461,.T.);
#463=ORIENTED_EDGE('', *,*, #313,.F.);
#464=ORIENTED_EDGE('', *,*, #97,.F.);
#465=CARTESIAN_POINT('', (0.031000000000000,-
0.183398473276088,0.617500000000000));
#466=VERTEX_POINT('', #465);
#467=DIRECTION('', (0.0,0.0,1.0));
#468=VECTOR('', #467,0.617500000000000);
#469=LINE('', #82,#468);
#470=EDGE_CURVE('', #84,#466,#469,.T.);
#471=ORIENTED_EDGE('', *,*, #470,.T.);

```



```

#472=CARTESIAN_POINT('', (0.0185000000000000, -
0.185077686391418, 0.6300000000000000));
#473=VERTEX_POINT('', #472);
#474=CARTESIAN_POINT('', (0.0185000000000000, -
0.185077686391418, 0.6300000000000000));
#475=CARTESIAN_POINT('', (0.020288448038622, -
0.184898916657693, 0.6300000000000000));
#476=CARTESIAN_POINT('', (0.023535786372608, -
0.184523602834953, 0.629279175846403));
#477=CARTESIAN_POINT('', (0.027638547132190, -
0.183946283585950, 0.626500455649787));
#478=CARTESIAN_POINT('', (0.030354567108582, -
0.183509179238234, 0.622337258170825));
#479=CARTESIAN_POINT('', (0.0310000000000000, -
0.183398473276088, 0.619203758810859));
#480=CARTESIAN_POINT('', (0.0310000000000000, -
0.183398473276088, 0.6175000000000000));
#481=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#474, #475, #476, #477, #478, #479, #480), .HYP
ERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 4), (0.0, 0.2500000000000000, 0.5000000000000000, 0.7
5000000000000000, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#482=EDGE_CURVE('', #466, #473, #481, .F.);
#483=ORIENTED_EDGE('', *, *, #482, .T.);
#484=CARTESIAN_POINT('', (-0.0185000000000000, -
0.185077686391418, 0.6300000000000000));
#485=VERTEX_POINT('', #484);
#486=CARTESIAN_POINT('', (0.0, 0.0, 0.6300000000000000));
#487=DIRECTION('', (0.0, 0.0, 1.0));
#488=DIRECTION('', (-0.099462365591398, -0.995041324685042, 0.0));
#489=AXIS2_PLACEMENT_3D('', #486, #487, #488);
#490=CIRCLE('', #489, 0.1860000000000000);
#491=EDGE_CURVE('', #473, #485, #490, .F.);
#492=ORIENTED_EDGE('', *, *, #491, .T.);
#493=CARTESIAN_POINT('', (-0.0310000000000000, -
0.183398473276088, 0.6175000000000000));
#494=VERTEX_POINT('', #493);
#495=CARTESIAN_POINT('', (-0.0310000000000000, -
0.183398473276088, 0.6175000000000000));
#496=CARTESIAN_POINT('', (-0.0310000000000000, -
0.183398473276088, 0.619276684284376));
#497=CARTESIAN_POINT('', (-0.030285994332659, -
0.183520653386331, 0.622509121211843));
#498=CARTESIAN_POINT('', (-0.027542467419886, -
0.183961116646316, 0.626594726403956));
#499=CARTESIAN_POINT('', (-0.023380441464035, -
0.184543239000241, 0.629340957979689));
#500=CARTESIAN_POINT('', (-0.020222806907318, -
0.184905478015536, 0.6300000000000000));
#501=CARTESIAN_POINT('', (-0.0185000000000000, -
0.185077686391418, 0.6300000000000000));
#502=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#495, #496, #497, #498, #499, #500, #501), .HYP
ERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 4), (0.0, 0.2500000000000000, 0.5000000000000000, 0.7
5000000000000000, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#503=EDGE_CURVE('', #485, #494, #502, .F.);

```

```

#504=ORIENTED_EDGE('',*,*,#503,.T.);
#505=DIRECTION('',(0.0,0.0,1.0));
#506=VECTOR('',#505,0.6175000000000000);
#507=LINE('',#44,#506);
#508=EDGE_CURVE('',#494,#46,#507,.F.);
#509=ORIENTED_EDGE('',*,*,#508,.T.);
#510=ORIENTED_EDGE('',*,*,#67,.F.);
#511=ORIENTED_EDGE('',*,*,#259,.F.);
#512=CARTESIAN_POINT('',(-0.160499221181911,-
0.0940000000000000,0.2500000000000000));
#513=VERTEX_POINT('',#512);
#514=CARTESIAN_POINT('',(-0.160499221181911,-
0.0940000000000000,0.2500000000000000));
#515=CARTESIAN_POINT('',(-0.160499221181911,-
0.0940000000000000,0.243237928218130));
#516=CARTESIAN_POINT('',(-0.161326264069271,-
0.092617528844491,0.229892455717941));
#517=CARTESIAN_POINT('',(-0.165188657251781,-
0.085684909394755,0.209422867545884));
#518=CARTESIAN_POINT('',(-0.170601713482259,-
0.074581303037190,0.191462225064981));
#519=CARTESIAN_POINT('',(-0.176608347816713,-
0.059230733088108,0.175968221892797));
#520=CARTESIAN_POINT('',(-0.181735072423795,-
0.041287079595952,0.164625925372114));
#521=CARTESIAN_POINT('',(-0.185280833739912,-
0.020378752353648,0.157404452422345));
#522=CARTESIAN_POINT('',(-0.1860000000000000,-
0.006823032072908,0.1560000000000000));
#523=CARTESIAN_POINT('',(-0.1860000000000000,2.277843E-17,0.1560000000000000));
#524=B_SPLINE_CURVE_WITH_KNOTS('',3,(#514,#515,#516,#517,#518,#519,#520,#521,
#522,#523),.HYPERBOLIC_ARC.,.F.,.F.,(4,1,1,1,1,1,1,4),(0.0,0.142857142857143,
0.285714285714286,0.428571428571429,0.571428571428571,0.714285714285714,0.857
142857142857,1.0),.PIECEWISE_BEZIER_KNOTS.);
#525=EDGE_CURVE('',#242,#513,#524,.F.);
#526=ORIENTED_EDGE('',*,*,#525,.T.);
#527=CARTESIAN_POINT('',(-0.1860000000000000,2.277843E-17,0.3440000000000001));
#528=CARTESIAN_POINT('',(-0.1860000000000000,-
0.006812818398979,0.3440000000000001));
#529=CARTESIAN_POINT('',(-0.185281410572974,-
0.020351551813128,0.342596407319869));
#530=CARTESIAN_POINT('',(-0.181755664328797,-
0.041193271352137,0.335416626725464));
#531=CARTESIAN_POINT('',(-0.176625064117123,-
0.059189393219094,0.324074232296005));
#532=CARTESIAN_POINT('',(-0.170594327073985,-
0.074600077742975,0.308518027938217));
#533=CARTESIAN_POINT('',(-0.165177934867762,-
0.085705567809687,0.290544495839190));
#534=CARTESIAN_POINT('',(-0.161309755247970,-
0.092646107527992,0.269982549598585));
#535=CARTESIAN_POINT('',(-0.160499221181911,-
0.0940000000000000,0.256708631292058));

```

```

#536=CARTESIAN_POINT('', (-0.160499221181911, -
0.0940000000000000, 0.2500000000000000));
#537=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#527, #528, #529, #530, #531, #532, #533, #534,
#535, #536), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 1, 4), (0.0, 0.142857142857143,
0.285714285714286, 0.428571428571429, 0.571428571428571, 0.714285714285714, 0.857
142857142857, 1.0), .PIECEWISE_BEZIER_KNOTS.);
#538=EDGE_CURVE('', #513, #220, #537, .F.);
#539=ORIENTED_EDGE('', *, *, #538, .T.);
#540=ORIENTED_EDGE('', *, *, #224, .F.);
#541=CARTESIAN_POINT('', (-0.175184902317523, -
0.0625000000000000, 2.5800000000000000));
#542=CARTESIAN_POINT('', (-0.175184902317523, -
0.0625000000000000, 2.574577143924821));
#543=CARTESIAN_POINT('', (-0.175676556620784, -
0.061146675813468, 2.564057738166918));
#544=CARTESIAN_POINT('', (-0.177772939484603, -
0.054932562783283, 2.548660674259625));
#545=CARTESIAN_POINT('', (-0.180647351963460, -
0.044822005907028, 2.535436160279311));
#546=CARTESIAN_POINT('', (-0.183444961303938, -
0.031895399078691, 2.525384006817258));
#547=CARTESIAN_POINT('', (-0.185536257371889, -
0.016107401839008, 2.518854091511808));
#548=CARTESIAN_POINT('', (-0.1860000000000000, -
0.005471621857183, 2.5175000000000000));
#549=CARTESIAN_POINT('', (-0.1860000000000000, 2.277843E-17, 2.5175000000000000));
#550=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#541, #542, #543, #544, #545, #546, #547, #548,
#549), .HYPERBOLIC_ARC., .F., .F., (4, 1, 1, 1, 1, 1, 1, 4), (0.0, 0.166666666666667, 0.33333
3333333333, 0.5000000000000000, 0.666666666666667, 0.833333333333333, 1.0), .PIECEW
ISE_BEZIER_KNOTS.);
#551=EDGE_CURVE('', #206, #377, #550, .F.);
#552=ORIENTED_EDGE('', *, *, #551, .T.);
#553=EDGE_LOOP('', (#389, #390, #399, #406, #407, #421, #433, #434, #449, #462, #463, #46
4, #471, #483, #492, #504, #509, #510, #511, #526, #539, #540, #552));
#554=FACE_BOUND('', #553, .F.);
#555=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#556=DIRECTION('', (0.0, 0.0, 1.0));
#557=DIRECTION('', (1.0, 0.0, 0.0));
#558=AXIS2_PLACEMENT_3D('', #555, #556, #557);
#559=CYLINDRICAL_SURFACE('', #558, 0.1860000000000000);
#560=ADVANCED_FACE('', (#554), #559, .T.);
#561=SHAPE_ASPECT('', '#43, .T.);
#562=PROPERTY_DEFINITION('', '#561);
#563=FACE_SHAPE_REPRESENTATION('', (#560), #34);
#564=SHAPE_DEFINITION_REPRESENTATION(#562, #563);
#565=ORIENTED_EDGE('', *, *, #51, .F.);
#566=ORIENTED_EDGE('', *, *, #508, .F.);
#567=DIRECTION('', (0.0, -1.0, 0.0));
#568=VECTOR('', #567, 0.366796946552176);
#569=LINE('', #262, #568);
#570=EDGE_CURVE('', #494, #263, #569, .F.);
#571=ORIENTED_EDGE('', *, *, #570, .T.);
#572=ORIENTED_EDGE('', *, *, #267, .F.);

```

```

#573=EDGE_LOOP('', (#565, #566, #571, #572));
#574=FACE_BOUND('', #573, .F.);
#575=CARTESIAN_POINT('', (-
0.0310000000000000, 0.0940000000000000, 0.2500000000000000));
#576=CARTESIAN_POINT('', (-0.0310000000000000, -
0.0940000000000000, 0.2500000000000000));
#577=VERTEX_POINT('', #575);
#578=VERTEX_POINT('', #576);
#579=CARTESIAN_POINT('', (-0.0310000000000000, 0.0, 0.2500000000000000));
#580=DIRECTION('', (-1.0, 0.0, 0.0));
#581=DIRECTION('', (0.0, -1.0, 0.0));
#582=AXIS2_PLACEMENT_3D('', #579, #580, #581);
#583=CIRCLE('', #582, 0.0940000000000000);
#584=EDGE_CURVE('', #577, #578, #583, .F.);
#585=ORIENTED_EDGE('', *, *, #584, .T.);
#586=CARTESIAN_POINT('', (-0.0310000000000000, 0.0, 0.2500000000000000));
#587=DIRECTION('', (-1.0, 0.0, 0.0));
#588=DIRECTION('', (0.0, 1.0, 0.0));
#589=AXIS2_PLACEMENT_3D('', #586, #587, #588);
#590=CIRCLE('', #589, 0.0940000000000000);
#591=EDGE_CURVE('', #578, #577, #590, .F.);
#592=ORIENTED_EDGE('', *, *, #591, .T.);
#593=EDGE_LOOP('', (#585, #592));
#594=FACE_BOUND('', #593, .F.);
#595=CARTESIAN_POINT('', (-0.0310000000000000, 0.1860000000000000, 0.0));
#596=DIRECTION('', (-1.0, 0.0, 0.0));
#597=DIRECTION('', (0.0, 0.0, 1.0));
#598=AXIS2_PLACEMENT_3D('', #595, #596, #597);
#599=PLANE('', #598);
#600=ADVANCED_FACE('', (#574, #594), #599, .F.);
#601=SHAPE_ASPECT('featured shape: face 6', '', #43, .T.);
#602=PROPERTY_DEFINITION('', '', #601);
#603=FACE_SHAPE_REPRESENTATION('', (#600), #34);
#604=SHAPE_DEFINITION_REPRESENTATION(#602, #603);
#605=ORIENTED_EDGE('', *, *, #279, .F.);
#606=ORIENTED_EDGE('', *, *, #570, .F.);
#607=ORIENTED_EDGE('', *, *, #503, .F.);
#608=DIRECTION('', (0.0, -1.0, 0.0));
#609=VECTOR('', #608, 0.370155372782836);
#610=LINE('', #269, #609);
#611=EDGE_CURVE('', #485, #270, #610, .F.);
#612=ORIENTED_EDGE('', *, *, #611, .T.);
#613=EDGE_LOOP('', (#605, #606, #607, #612));
#614=FACE_BOUND('', #613, .F.);
#615=CARTESIAN_POINT('', (-
0.0185000000000000, 0.1860000000000000, 0.6175000000000000));
#616=DIRECTION('', (0.0, 1.0, 0.0));
#617=DIRECTION('', (-1.0, 0.0, 0.0));
#618=AXIS2_PLACEMENT_3D('', #615, #616, #617);
#619=CYLINDRICAL_SURFACE('', #618, 0.0125000000000000);
#620=ADVANCED_FACE('', (#614), #619, .F.);
#621=SHAPE_ASPECT('featured shape: face 7', '', #43, .T.);
#622=PROPERTY_DEFINITION('', '', #621);

```

```

#623=FACE_SHAPE_REPRESENTATION('', (#620), #34);
#624=SHAPE_DEFINITION_REPRESENTATION(#622, #623);
#625=ORIENTED_EDGE('', *, *, #288, .F.);
#626=ORIENTED_EDGE('', *, *, #611, .F.);
#627=ORIENTED_EDGE('', *, *, #491, .F.);
#628=DIRECTION('', (0.0, -1.0, 0.0));
#629=VECTOR('', #628, 0.370155372782836);
#630=LINE('', #281, #629);
#631=EDGE_CURVE('', #473, #282, #630, .F.);
#632=ORIENTED_EDGE('', *, *, #631, .T.);
#633=EDGE_LOOP('', (#625, #626, #627, #632));
#634=FACE_BOUND('', #633, .F.);
#635=CARTESIAN_POINT('', (-
0.0185000000000000, 0.1860000000000000, 0.6300000000000000));
#636=DIRECTION('', (0.0, 0.0, 1.0));
#637=DIRECTION('', (1.0, 0.0, 0.0));
#638=AXIS2_PLACEMENT_3D('', #635, #636, #637);
#639=PLANE('', #638);
#640=ADVANCED_FACE('', (#634), #639, .F.);
#641=SHAPE_ASPECT('featured shape: face 8', '', #43, .T.);
#642=PROPERTY_DEFINITION('', '', #641);
#643=FACE_SHAPE_REPRESENTATION('', (#640), #34);
#644=SHAPE_DEFINITION_REPRESENTATION(#642, #643);
#645=ORIENTED_EDGE('', *, *, #300, .F.);
#646=ORIENTED_EDGE('', *, *, #631, .F.);
#647=ORIENTED_EDGE('', *, *, #482, .F.);
#648=DIRECTION('', (0.0, -1.0, 0.0));
#649=VECTOR('', #648, 0.366796946552176);
#650=LINE('', #290, #649);
#651=EDGE_CURVE('', #466, #291, #650, .F.);
#652=ORIENTED_EDGE('', *, *, #651, .T.);
#653=EDGE_LOOP('', (#645, #646, #647, #652));
#654=FACE_BOUND('', #653, .F.);
#655=CARTESIAN_POINT('', (0.0185000000000000, 0.1860000000000000, 0.6175000000000000));
#656=DIRECTION('', (0.0, 1.0, 0.0));
#657=DIRECTION('', (-1.0, 0.0, 0.0));
#658=AXIS2_PLACEMENT_3D('', #655, #656, #657);
#659=CYLINDRICAL_SURFACE('', #658, 0.0125000000000000);
#660=ADVANCED_FACE('', (#654), #659, .F.);
#661=SHAPE_ASPECT('featured shape: face 9', '', #43, .T.);
#662=PROPERTY_DEFINITION('', '', #661);
#663=FACE_SHAPE_REPRESENTATION('', (#660), #34);
#664=SHAPE_DEFINITION_REPRESENTATION(#662, #663);
#665=ORIENTED_EDGE('', *, *, #88, .F.);
#666=ORIENTED_EDGE('', *, *, #305, .F.);
#667=ORIENTED_EDGE('', *, *, #651, .F.);
#668=ORIENTED_EDGE('', *, *, #470, .F.);
#669=EDGE_LOOP('', (#665, #666, #667, #668));
#670=FACE_BOUND('', #669, .F.);
#671=CARTESIAN_POINT('', (0.0310000000000000, -
0.0940000000000000, 0.2500000000000000));

```

```

#672=CARTESIAN_POINT('', (0.0310000000000000, 0.0940000000000000, 0.2500000000000000
0));
#673=VERTEX_POINT('', #671);
#674=VERTEX_POINT('', #672);
#675=CARTESIAN_POINT('', (0.0310000000000000, 0.0, 0.2500000000000000));
#676=DIRECTION('', (-1.0, 0.0, 0.0));
#677=DIRECTION('', (0.0, -1.0, 0.0));
#678=AXIS2_PLACEMENT_3D('', #675, #676, #677);
#679=CIRCLE('', #678, 0.0940000000000000);
#680=EDGE_CURVE('', #673, #674, #679, .T.);
#681=ORIENTED_EDGE('', *, *, #680, .T.);
#682=CARTESIAN_POINT('', (0.0310000000000000, 0.0, 0.2500000000000000));
#683=DIRECTION('', (-1.0, 0.0, 0.0));
#684=DIRECTION('', (0.0, 1.0, 0.0));
#685=AXIS2_PLACEMENT_3D('', #682, #683, #684);
#686=CIRCLE('', #685, 0.0940000000000000);
#687=EDGE_CURVE('', #674, #673, #686, .T.);
#688=ORIENTED_EDGE('', *, *, #687, .T.);
#689=EDGE_LOOP('', (#681, #688));
#690=FACE_BOUND('', #689, .F.);
#691=CARTESIAN_POINT('', (0.0310000000000000, 0.1860000000000000, 0.6175000000000000
0));
#692=DIRECTION('', (1.0, 0.0, 0.0));
#693=DIRECTION('', (0.0, 0.0, -1.0));
#694=AXIS2_PLACEMENT_3D('', #691, #692, #693);
#695=PLANE('', #694);
#696=ADVANCED_FACE('', (#670, #690), #695, .F.);
#697=SHAPE_ASPECT('featured shape: face 10', '', #43, .T.);
#698=PROPERTY_DEFINITION('', '', #697);
#699=FACE_SHAPE_REPRESENTATION('', (#696), #34);
#700=SHAPE_DEFINITION_REPRESENTATION(#698, #699);
#701=ORIENTED_EDGE('', *, *, #343, .F.);
#702=DIRECTION('', (-1.0, 0.0, 0.0));
#703=VECTOR('', #702, 0.129499221181911);
#704=LINE('', #315, #703);
#705=EDGE_CURVE('', #316, #674, #704, .T.);
#706=ORIENTED_EDGE('', *, *, #705, .T.);
#707=ORIENTED_EDGE('', *, *, #680, .F.);
#708=DIRECTION('', (-1.0, 0.0, 0.0));
#709=VECTOR('', #708, 0.129499221181911);
#710=LINE('', #435, #709);
#711=EDGE_CURVE('', #673, #436, #710, .F.);
#712=ORIENTED_EDGE('', *, *, #711, .T.);
#713=ORIENTED_EDGE('', *, *, #448, .F.);
#714=EDGE_LOOP('', (#701, #706, #707, #712, #713));
#715=FACE_BOUND('', #714, .F.);
#716=CARTESIAN_POINT('', (0.1860000000000000, 0.0, 0.2500000000000000));
#717=DIRECTION('', (-1.0, 0.0, 0.0));
#718=DIRECTION('', (0.0, -1.0, 0.0));
#719=AXIS2_PLACEMENT_3D('', #716, #717, #718);
#720=CYLINDRICAL_SURFACE('', #719, 0.0940000000000000);
#721=ADVANCED_FACE('', (#715), #720, .F.);
#722=SHAPE_ASPECT('featured shape: face 11', '', #43, .T.);

```

```

#723=PROPERTY_DEFINITION('', '#722');
#724=FACE_SHAPE_REPRESENTATION('', (#721), #34);
#725=SHAPE_DEFINITION_REPRESENTATION(#723, #724);
#726=ORIENTED_EDGE('', *, *, #239, .F.);
#727=ORIENTED_EDGE('', *, *, #538, .F.);
#728=DIRECTION('', (-1.0, 0.0, 0.0));
#729=VECTOR('', #728, 0.129499221181911);
#730=LINE('', #576, #729);
#731=EDGE_CURVE('', #513, #578, #730, .F.);
#732=ORIENTED_EDGE('', *, *, #731, .T.);
#733=ORIENTED_EDGE('', *, *, #584, .F.);
#734=DIRECTION('', (-1.0, 0.0, 0.0));
#735=VECTOR('', #734, 0.129499221181911);
#736=LINE('', #575, #735);
#737=EDGE_CURVE('', #577, #227, #736, .T.);
#738=ORIENTED_EDGE('', *, *, #737, .T.);
#739=EDGE_LOOP('', (#726, #727, #732, #733, #738));
#740=FACE_BOUND('', #739, .F.);
#741=ADVANCED_FACE('', (#740), #720, .F.);
#742=SHAPE_ASPECT('featured shape: face 12', '', #43, .T.);
#743=PROPERTY_DEFINITION('', '#742');
#744=FACE_SHAPE_REPRESENTATION('', (#741), #34);
#745=SHAPE_DEFINITION_REPRESENTATION(#743, #744);
#746=ORIENTED_EDGE('', *, *, #254, .F.);
#747=ORIENTED_EDGE('', *, *, #737, .F.);
#748=ORIENTED_EDGE('', *, *, #591, .F.);
#749=ORIENTED_EDGE('', *, *, #731, .F.);
#750=ORIENTED_EDGE('', *, *, #525, .F.);
#751=EDGE_LOOP('', (#746, #747, #748, #749, #750));
#752=FACE_BOUND('', #751, .F.);
#753=CARTESIAN_POINT('', (0.186000000000000, 0.0, 0.250000000000000));
#754=DIRECTION('', (-1.0, 0.0, 0.0));
#755=DIRECTION('', (0.0, -1.0, 0.0));
#756=AXIS2_PLACEMENT_3D('', #753, #754, #755);
#757=CYLINDRICAL_SURFACE('', #756, 0.094000000000000);
#758=ADVANCED_FACE('', (#752), #757, .F.);
#759=SHAPE_ASPECT('featured shape: face 13', '', #43, .T.);
#760=PROPERTY_DEFINITION('', '#759');
#761=FACE_SHAPE_REPRESENTATION('', (#758), #34);
#762=SHAPE_DEFINITION_REPRESENTATION(#760, #761);
#763=ORIENTED_EDGE('', *, *, #328, .F.);
#764=ORIENTED_EDGE('', *, *, #461, .F.);
#765=ORIENTED_EDGE('', *, *, #711, .F.);
#766=ORIENTED_EDGE('', *, *, #687, .F.);
#767=ORIENTED_EDGE('', *, *, #705, .F.);
#768=EDGE_LOOP('', (#763, #764, #765, #766, #767));
#769=FACE_BOUND('', #768, .F.);
#770=ADVANCED_FACE('', (#769), #757, .F.);
#771=SHAPE_ASPECT('featured shape: face 14', '', #43, .T.);
#772=PROPERTY_DEFINITION('', '#771');
#773=FACE_SHAPE_REPRESENTATION('', (#770), #34);
#774=SHAPE_DEFINITION_REPRESENTATION(#772, #773);
#775=CARTESIAN_POINT('', (-7.654042E-18, 0.125000000000000, 3.94000000000000));

```

```

#776=CARTESIAN_POINT('', (-7.654042E-18, -
0.1250000000000000, 3.940000000000000));
#777=VERTEX_POINT('', #775);
#778=VERTEX_POINT('', #776);
#779=CARTESIAN_POINT('', (0.0, 0.0, 3.940000000000000));
#780=DIRECTION('', (0.0, 0.0, 1.0));
#781=DIRECTION('', (0.0, -1.0, 0.0));
#782=AXIS2_PLACEMENT_3D('', #779, #780, #781);
#783=CIRCLE('', #782, 0.1250000000000000);
#784=EDGE_CURVE('', #777, #778, #783, .F.);
#785=ORIENTED_EDGE('', *, *, #784, .T.);
#786=CARTESIAN_POINT('', (0.0, -0.1250000000000000, 4.179999999999999));
#787=VERTEX_POINT('', #786);
#788=DIRECTION('', (0.0, 0.0, -1.0));
#789=VECTOR('', #788, 0.2400000000000001);
#790=LINE('', #786, #789);
#791=EDGE_CURVE('', #778, #787, #790, .F.);
#792=ORIENTED_EDGE('', *, *, #791, .T.);
#793=CARTESIAN_POINT('', (-1.530808E-17, 0.1250000000000000, 4.180000000000000));
#794=VERTEX_POINT('', #793);
#795=CARTESIAN_POINT('', (0.0, 0.0, 4.180000000000000));
#796=DIRECTION('', (0.0, 0.0, 1.0));
#797=DIRECTION('', (0.0, -1.0, 0.0));
#798=AXIS2_PLACEMENT_3D('', #795, #796, #797);
#799=CIRCLE('', #798, 0.1250000000000000);
#800=EDGE_CURVE('', #787, #794, #799, .T.);
#801=ORIENTED_EDGE('', *, *, #800, .T.);
#802=DIRECTION('', (0.0, 0.0, -1.0));
#803=VECTOR('', #802, 0.2400000000000000);
#804=LINE('', #793, #803);
#805=EDGE_CURVE('', #794, #777, #804, .T.);
#806=ORIENTED_EDGE('', *, *, #805, .T.);
#807=EDGE_LOOP('', (#785, #792, #801, #806));
#808=FACE_BOUND('', #807, .F.);
#809=CARTESIAN_POINT('', (0.0, 0.0, 3.921499999999999));
#810=DIRECTION('', (0.0, 0.0, 1.0));
#811=DIRECTION('', (0.0, -1.0, 0.0));
#812=AXIS2_PLACEMENT_3D('', #809, #810, #811);
#813=CYLINDRICAL_SURFACE('', #812, 0.1250000000000000);
#814=ADVANCED_FACE('', (#808), #813, .T.);
#815=SHAPE_ASPECT('featured shape: face 15', '', #43, .T.);
#816=PROPERTY_DEFINITION('', '', #815);
#817=FACE_SHAPE_REPRESENTATION('', (#814), #34);
#818=SHAPE_DEFINITION_REPRESENTATION(#816, #817);
#819=CARTESIAN_POINT('', (0.0, 0.0, 3.939999999999999));
#820=DIRECTION('', (0.0, 0.0, -1.0));
#821=DIRECTION('', (0.0, -1.0, 0.0));
#822=AXIS2_PLACEMENT_3D('', #819, #820, #821);
#823=CIRCLE('', #822, 0.1250000000000000);
#824=EDGE_CURVE('', #778, #777, #823, .T.);
#825=ORIENTED_EDGE('', *, *, #824, .T.);
#826=ORIENTED_EDGE('', *, *, #805, .F.);
#827=CARTESIAN_POINT('', (0.0, 0.0, 4.180000000000000));

```



```

#828=DIRECTION('', (0.0,0.0,1.0));
#829=DIRECTION('', (-1.224647E-16,1.0,0.0));
#830=AXIS2_PLACEMENT_3D('', #827, #828, #829);
#831=CIRCLE('', #830, 0.1250000000000000);
#832=EDGE_CURVE('', #794, #787, #831, .T.);
#833=ORIENTED_EDGE('', *, *, #832, .T.);
#834=ORIENTED_EDGE('', *, *, #791, .F.);
#835=EDGE_LOOP('', (#825, #826, #833, #834));
#836=FACE_BOUND('', #835, .F.);
#837=CARTESIAN_POINT('', (0.0,0.0,3.921499999999999));
#838=DIRECTION('', (0.0,0.0,1.0));
#839=DIRECTION('', (0.0,-1.0,0.0));
#840=AXIS2_PLACEMENT_3D('', #837, #838, #839);
#841=CYLINDRICAL_SURFACE('', #840, 0.1250000000000000);
#842=ADVANCED_FACE('', (#836), #841, .T.);
#843=SHAPE_ASPECT('featured shape: face 16', '', #43, .T.);
#844=PROPERTY_DEFINITION('', '', #843);
#845=FACE_SHAPE_REPRESENTATION('', (#842), #34);
#846=SHAPE_DEFINITION_REPRESENTATION(#844, #845);
#847=ORIENTED_EDGE('', *, *, #122, .F.);
#848=DIRECTION('', (-3.169619E-17,0.258819045102521,-0.965925826289068));
#849=VECTOR('', #848, 0.134585903453311);
#850=LINE('', #113, #849);
#851=EDGE_CURVE('', #115, #794, #850, .T.);
#852=ORIENTED_EDGE('', *, *, #851, .T.);
#853=ORIENTED_EDGE('', *, *, #800, .F.);
#854=DIRECTION('', (0.0,-0.258819045102521,-0.965925826289068));
#855=VECTOR('', #854, 0.134585903453311);
#856=LINE('', #114, #855);
#857=EDGE_CURVE('', #787, #116, #856, .F.);
#858=ORIENTED_EDGE('', *, *, #857, .T.);
#859=EDGE_LOOP('', (#847, #852, #853, #858));
#860=FACE_BOUND('', #859, .F.);
#861=CARTESIAN_POINT('', (0.0,0.0,4.180000000000000));
#862=DIRECTION('', (0.0,0.0,-1.0));
#863=DIRECTION('', (0.0,1.0,0.0));
#864=AXIS2_PLACEMENT_3D('', #861, #862, #863);
#865=CONICAL_SURFACE('', #864, 0.1250000000000000, 15.000000000000009);
#866=ADVANCED_FACE('', (#860), #865, .T.);
#867=SHAPE_ASPECT('featured shape: face 17', '', #43, .T.);
#868=PROPERTY_DEFINITION('', '', #867);
#869=FACE_SHAPE_REPRESENTATION('', (#866), #34);
#870=SHAPE_DEFINITION_REPRESENTATION(#868, #869);
#871=ORIENTED_EDGE('', *, *, #129, .F.);
#872=ORIENTED_EDGE('', *, *, #857, .F.);
#873=ORIENTED_EDGE('', *, *, #832, .F.);
#874=ORIENTED_EDGE('', *, *, #851, .F.);
#875=EDGE_LOOP('', (#871, #872, #873, #874));
#876=FACE_BOUND('', #875, .F.);
#877=CARTESIAN_POINT('', (0.0,0.0,4.180000000000000));
#878=DIRECTION('', (0.0,0.0,-1.0));
#879=DIRECTION('', (0.0,1.0,0.0));
#880=AXIS2_PLACEMENT_3D('', #877, #878, #879);

```

```

#881=CONICAL_SURFACE('',#880,0.125000000000000,15.000000000000009);
#882=ADVANCED_FACE('',(#876),#881,.T.);
#883=SHAPE_ASPECT('featured shape: face 18','',#43,.T.);
#884=PROPERTY_DEFINITION('',',',#883);
#885=FACE_SHAPE_REPRESENTATION('',(#882),#34);
#886=SHAPE_DEFINITION_REPRESENTATION(#884,#885);
#887=ORIENTED_EDGE('',*,*,#173,.F.);
#888=ORIENTED_EDGE('',*,*,#405,.F.);
#889=DIRECTION('',(4.891782E-17,0.399444320513099,0.916757456915311));
#890=VECTOR('',#889,0.152712147519443);
#891=LINE('',#391,#890);
#892=EDGE_CURVE('',#392,#778,#891,.T.);
#893=ORIENTED_EDGE('',*,*,#892,.T.);
#894=ORIENTED_EDGE('',*,*,#784,.F.);
#895=DIRECTION('',(0.0,-0.399444320513099,0.916757456915311));
#896=VECTOR('',#895,0.152712147519443);
#897=LINE('',#166,#896);
#898=EDGE_CURVE('',#777,#167,#897,.F.);
#899=ORIENTED_EDGE('',*,*,#898,.T.);
#900=EDGE_LOOP('',(#887,#888,#893,#894,#899));
#901=FACE_BOUND('',#900,.F.);
#902=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#903=DIRECTION('',(0.0,0.0,-1.0));
#904=DIRECTION('',(0.0,1.0,0.0));
#905=AXIS2_PLACEMENT_3D('',#902,#903,#904);
#906=CONICAL_SURFACE('',#905,0.186000000000000,23.543444867436634);
#907=ADVANCED_FACE('',(#901),#906,.T.);
#908=SHAPE_ASPECT('featured shape: face 19','',#43,.T.);
#909=PROPERTY_DEFINITION('',',',#908);
#910=FACE_SHAPE_REPRESENTATION('',(#907),#34);
#911=SHAPE_DEFINITION_REPRESENTATION(#909,#910);
#912=ORIENTED_EDGE('',*,*,#182,.F.);
#913=ORIENTED_EDGE('',*,*,#898,.F.);
#914=ORIENTED_EDGE('',*,*,#824,.F.);
#915=ORIENTED_EDGE('',*,*,#892,.F.);
#916=ORIENTED_EDGE('',*,*,#398,.F.);
#917=EDGE_LOOP('',(#912,#913,#914,#915,#916));
#918=FACE_BOUND('',#917,.F.);
#919=CARTESIAN_POINT('',(0.0,0.0,3.799999999999999));
#920=DIRECTION('',(0.0,0.0,-1.0));
#921=DIRECTION('',(0.0,1.0,0.0));
#922=AXIS2_PLACEMENT_3D('',#919,#920,#921);
#923=CONICAL_SURFACE('',#922,0.186000000000000,23.543444867436634);
#924=ADVANCED_FACE('',(#918),#923,.T.);
#925=SHAPE_ASPECT('featured shape: face 20','',#43,.T.);
#926=PROPERTY_DEFINITION('',',',#925);
#927=FACE_SHAPE_REPRESENTATION('',(#924),#34);
#928=SHAPE_DEFINITION_REPRESENTATION(#926,#927);
#929=ORIENTED_EDGE('',*,*,#157,.F.);
#930=DIRECTION('',(-1.0,0.0,0.0));
#931=VECTOR('',#930,0.350369804635046);
#932=LINE('',#143,#931);
#933=EDGE_CURVE('',#145,#192,#932,.T.);

```

```

#934=ORIENTED_EDGE('',*,*,#933,.T.);
#935=ORIENTED_EDGE('',*,*,#203,.F.);
#936=ORIENTED_EDGE('',*,*,#388,.F.);
#937=DIRECTION('',(-1.0,0.0,0.0));
#938=VECTOR('',#937,0.350369804635046);
#939=LINE('',#408,#938);
#940=EDGE_CURVE('',#377,#409,#939,.F.);
#941=ORIENTED_EDGE('',*,*,#940,.T.);
#942=ORIENTED_EDGE('',*,*,#420,.F.);
#943=EDGE_LOOP('',(#929,#934,#935,#936,#941,#942));
#944=FACE_BOUND('',#943,.F.);
#945=CARTESIAN_POINT('',(0.186000000000000,0.0,2.58000000000000));
#946=DIRECTION('',(-1.0,0.0,0.0));
#947=DIRECTION('',(0.0,-1.0,0.0));
#948=AXIS2_PLACEMENT_3D('',#945,#946,#947);
#949=CYLINDRICAL_SURFACE('',#948,0.062500000000000);
#950=ADVANCED_FACE('',(#944),#949,.F.);
#951=SHAPE_ASPECT('featured shape: face 21','',#43,.T.);
#952=PROPERTY_DEFINITION('',',',#951);
#953=FACE_SHAPE_REPRESENTATION('',(#950),#34);
#954=SHAPE_DEFINITION_REPRESENTATION(#952,#953);
#955=ORIENTED_EDGE('',*,*,#217,.F.);
#956=ORIENTED_EDGE('',*,*,#933,.F.);
#957=ORIENTED_EDGE('',*,*,#362,.F.);
#958=ORIENTED_EDGE('',*,*,#432,.F.);
#959=ORIENTED_EDGE('',*,*,#940,.F.);
#960=ORIENTED_EDGE('',*,*,#551,.F.);
#961=EDGE_LOOP('',(#955,#956,#957,#958,#959,#960));
#962=FACE_BOUND('',#961,.F.);
#963=CARTESIAN_POINT('',(0.186000000000000,0.0,2.58000000000000));
#964=DIRECTION('',(-1.0,0.0,0.0));
#965=DIRECTION('',(0.0,-1.0,0.0));
#966=AXIS2_PLACEMENT_3D('',#963,#964,#965);
#967=CYLINDRICAL_SURFACE('',#966,0.062500000000000);
#968=ADVANCED_FACE('',(#962),#967,.F.);
#969=SHAPE_ASPECT('featured shape: face 22','',#43,.T.);
#970=PROPERTY_DEFINITION('',',',#969);
#971=FACE_SHAPE_REPRESENTATION('',(#968),#34);
#972=SHAPE_DEFINITION_REPRESENTATION(#970,#971);
#973=CLOSED_SHELL('',(#76,#108,#138,#371,#560,#600,#620,#640,#660,#696,#721,#
741,#758,#770,#814,#842,#866,#882,#907,#924,#950,#968));
#974=MANIFOLD_SOLID_BREP('',#973);
#975=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#974),#34);
#976=SHAPE_DEFINITION_REPRESENTATION(#43,#975);
#977=CARTESIAN_POINT('',(0.0,0.0,0.0));
#978=DIRECTION('',(0.0,1.0,0.0));
#979=DIRECTION('',(0.0,0.0,1.0));
#980=AXIS2_PLACEMENT_3D('DTM1',#977,#978,#979);
#981=DATUM_FEATURE('',',',#43,.T.);
#982=PROPERTY_DEFINITION('',',',#981);
#983=SHAPE_REPRESENTATION('',(#980),#34);
#984=SHAPE_DEFINITION_REPRESENTATION(#982,#983);
#985=DATUM('DTM1','',#43,.F.,'DTM1');

```

```

#986=SHAPE_ASPECT_RELATIONSHIP('', '#981', '#985');
#987=DATUM_REFERENCE(1, '#985');
#988=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#989=DIRECTION('', (-1.0, 0.0, 0.0));
#990=DIRECTION('', (0.0, 0.0, 1.0));
#991=AXIS2_PLACEMENT_3D('DTM2', '#988', '#989', '#990');
#992=DATUM_FEATURE('', '#43', '.T. ');
#993=PROPERTY_DEFINITION('', '#992');
#994=SHAPE_REPRESENTATION('', (#991), #34);
#995=SHAPE_DEFINITION_REPRESENTATION(#993, #994);
#996=DATUM('DTM2', '#43', '.F.', 'DTM2');
#997=SHAPE_ASPECT_RELATIONSHIP('', '#992', '#996');
#998=DATUM_REFERENCE(1, '#996');
#999=CARTESIAN_POINT('', (0.186000000000000, 0.0, 2.155000000000000));
#1000=DIRECTION('', (-1.0, 0.0, 0.0));
#1001=DIRECTION('', (0.0, -1.0, 0.0));
#1002=AXIS2_PLACEMENT_3D('DTM3', '#999', '#1000', '#1001');
#1003=DATUM_FEATURE('', '#43', '.T. ');
#1004=PROPERTY_DEFINITION('', '#1003');
#1005=SHAPE_REPRESENTATION('', (#1002), #34);
#1006=SHAPE_DEFINITION_REPRESENTATION(#1004, #1005);
#1007=DATUM('DTM3', '#43', '.F.', 'DTM3');
#1008=SHAPE_ASPECT_RELATIONSHIP('', '#1003', '#1007');
#1009=DATUM_REFERENCE(1, '#1007');
#1010=CARTESIAN_POINT('', (0.0, 0.186000000000000, 2.155000000000000));
#1011=DIRECTION('', (0.0, 1.0, 0.0));
#1012=DIRECTION('', (1.0, 0.0, 0.0));
#1013=AXIS2_PLACEMENT_3D('DTM4', '#1010', '#1011', '#1012');
#1014=DATUM_FEATURE('', '#43', '.T. ');
#1015=PROPERTY_DEFINITION('', '#1014');
#1016=SHAPE_REPRESENTATION('', (#1013), #34);
#1017=SHAPE_DEFINITION_REPRESENTATION(#1015, #1016);
#1018=DATUM('DTM4', '#43', '.F.', 'DTM4');
#1019=SHAPE_ASPECT_RELATIONSHIP('', '#1014', '#1018');
#1020=DATUM_REFERENCE(1, '#1018');
#1021=CARTESIAN_POINT('', (0.186000000000000, 0.0, 0.250000000000000));
#1022=DIRECTION('', (0.0, 0.0, -1.0));
#1023=DIRECTION('', (-1.0, 0.0, 0.0));
#1024=AXIS2_PLACEMENT_3D('DTM5', '#1021', '#1022', '#1023');
#1025=DATUM_FEATURE('', '#43', '.T. ');
#1026=PROPERTY_DEFINITION('', '#1025');
#1027=SHAPE_REPRESENTATION('', (#1024), #34);
#1028=SHAPE_DEFINITION_REPRESENTATION(#1026, #1027);
#1029=DATUM('DTM5', '#43', '.T.', 'DTM5');
#1030=SHAPE_ASPECT_RELATIONSHIP('', '#1025', '#1029');
#1031=DATUM_REFERENCE(1, '#1029');
#1032=SHAPE_ASPECT('', '#43', '.T. ');

#1061=(CHARACTERIZED_OBJECT('', '#18143', '.T. ') FEATURE_DEFINITION() INSTANCED_FEATURE() SHAP
E_ASPECT('', '#18143', '.T. ') SLOT());

```

```

#1062=PROPERTY_DEFINITION('', '#1061');
#1063=PRODUCT_DEFINITION_SHAPE('', '#1061');
#1064=CARTESIAN_POINT('', (1.040834E-17, -
0.1860000000000000, 0.6300000000000000));
#1065=DIRECTION('', (0.0, 1.0, 0.0));
#1066=DIRECTION('', (1.0, 0.0, 0.0));
#1067=AXIS2_PLACEMENT_3D('orientation', #1064, #1065, #1066);

#1077=SHAPE_REPRESENTATION('', (#600, #620, #640, #660, #696), #34);
#1078=SHAPE_DEFINITION_REPRESENTATION(#1062, #1077);
#1079=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1067), #34);
#1080=SHAPE_DEFINITION_REPRESENTATION(#1062, #1079);
#1081=FEATURE_COMPONENT_DEFINITION('', '');
#1082=PRODUCT_DEFINITION_SHAPE('', '#1081');

#1083=SQUARE_U_PROFILE('', 'swept shape', #1082, .F.);
#1084=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#1085=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#1086=PRECISION_QUALIFIER(4);
#1092=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.0620000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1084
, #1085, #1086)) REPRESENTATION_ITEM('width'));
#1097=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #24) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('first angle'));
#1102=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #24) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('second angle'));
#1103=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0075000000000000);
#1104=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0075000000000000);
#1105=PRECISION_QUALIFIER(4);
#1111=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.0125000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1103
, #1104, #1105)) REPRESENTATION_ITEM('first radius'));
#1112=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0075000000000000);
#1113=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0075000000000000);
#1114=PRECISION_QUALIFIER(4);
#1120=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.0125000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1112
, #1113, #1114)) REPRESENTATION_ITEM('second radius'));
#1121=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1067, #1092, #1097, #1102, #1111,
#1120), #34);
#1122=PROPERTY_DEFINITION('', '#1083');
#1123=SHAPE_DEFINITION_REPRESENTATION(#1122, #1121);
#1124=SHAPE_ASPECT('', 'swept shape occurrence', #1063, .T.);
#1125=SHAPE_DEFINING_RELATIONSHIP('', 'profile usage', #1083, #1124);
#1126=FEATURE_COMPONENT_DEFINITION('', '');
#1127=PRODUCT_DEFINITION_SHAPE('', '#1126');

#1128=PATH_FEATURE_COMPONENT('linear path', 'linear', #1127, .F.);
#1133=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.3720000000000000), #15) REPRESENTATION_ITEM('distance'));
#1134=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1067, #1133), #34);
#1135=PROPERTY_DEFINITION('', '#1128');

```

```

#1136=SHAPE_DEFINITION_REPRESENTATION(#1135,#1134);
#1137=DIRECTION_SHAPE_REPRESENTATION('',(#1065),#34);
#1138=PROPERTY_DEFINITION_REPRESENTATION(#1135,#1137);
#1139=SHAPE_ASPECT('','course of travel occurrence',#1063,.T.);
#1140=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#1128,#1139);
#1141=FEATURE_COMPONENT_DEFINITION('', '');
#1142=PRODUCT_DEFINITION_SHAPE('', '#1141);
#1143=SLOT_END('', 'open', #1142, .F.);
#1144=FEATURE_COMPONENT_DEFINITION('', '');
#1145=PRODUCT_DEFINITION_SHAPE('', '#1144);
#1146=SLOT_END('', 'open', #1145, .F.);
#1147=SHAPE_ASPECT('', 'end condition occurrence', #1063, .T.);
#1148=FEATURE_COMPONENT_RELATIONSHIP('course of travel start', 'slot end
usage', #1143, #1147);
#1149=SHAPE_ASPECT('', 'end condition occurrence', #1063, .T.);
#1150=FEATURE_COMPONENT_RELATIONSHIP('course of travel end', 'slot end
usage', #1146, #1149);

#1157=(CHARACTERIZED_OBJECT('', '') FEATURE_DEFINITION() INSTANCED_FEATURE() ROUN
D_HOLE() SHAPE_ASPECT('', '#43, .T.));
#1158=PROPERTY_DEFINITION('', '#1157);
#1159=PRODUCT_DEFINITION_SHAPE('', '#1157);
#1160=CARTESIAN_POINT('', (0.186000000000000, 1.138922E-17, 0.250000000000000));
#1161=DIRECTION('', (-1.0, 0.0, 0.0));
#1162=DIRECTION('', (0.0, -1.0, 0.0));
#1163=AXIS2_PLACEMENT_3D('orientation', #1160, #1161, #1162);
#1164=SHAPE_REPRESENTATION('', (#721, #741, #758, #770), #34);
#1165=SHAPE_DEFINITION_REPRESENTATION(#1158, #1164);
#1166=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1163), #34);
#1167=SHAPE_DEFINITION_REPRESENTATION(#1158, #1166);
#1168=FEATURE_COMPONENT_DEFINITION('', '');
#1169=PRODUCT_DEFINITION_SHAPE('', '#1168);
#1170=CIRCULAR_CLOSED_PROFILE('circular profile', '#1169, .F.);
#1171=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.010000000000000);
#1172=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.010000000000000);
#1173=PRECISION_QUALIFIER(4);
#1179=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.188000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1171
, #1172, #1173) REPRESENTATION_ITEM('diameter'));
#1180=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1163, #1179), #34);
#1181=PROPERTY_DEFINITION('', '#1170);
#1182=SHAPE_DEFINITION_REPRESENTATION(#1181, #1180);
#1183=FEATURE_COMPONENT_DEFINITION('', '');
#1184=PRODUCT_DEFINITION_SHAPE('', '#1183);
#1185=PATH_FEATURE_COMPONENT('linear path', 'linear', #1184, .F.);
#1190=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.372000000000000), #15) REPRESENTATION_ITEM('distance'));
#1191=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1163, #1190), #34);
#1192=PROPERTY_DEFINITION('', '#1185);
#1193=SHAPE_DEFINITION_REPRESENTATION(#1192, #1191);
#1194=DIRECTION_SHAPE_REPRESENTATION('', (#1161), #34);
#1195=PROPERTY_DEFINITION_REPRESENTATION(#1192, #1194);

```

```

#1196=SHAPE_ASPECT('', 'diameter occurrence', #1159, .T.);
#1197=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #1170, #1196);
#1198=SHAPE_ASPECT('', 'hole depth occurrence', #1159, .T.);
#1199=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #1185, #1198);

#1200=FEATURE_COMPONENT_DEFINITION('', '');
#1201=PRODUCT_DEFINITION_SHAPE('', '', #1200);
#1202=HOLE_BOTTOM('bottom condition', 'through', #1201, .F.);
#1203=SHAPE_ASPECT('', 'bottom condition occurrence', #1159, .T.);
#1204=FEATURE_COMPONENT_RELATIONSHIP('', 'hole bottom usage', #1202, #1203);
#1205=CARTESIAN_POINT('', (0.1860000000000000, 1.387779E-17, 0.2500000000000000));
#1206=DIRECTION('', (-1.0, 0.0, 0.0));
#1207=AXIS1_PLACEMENT('', #1205, #1206);
#1208=SHAPE_ASPECT('', '', #43, .F.);
#1209=PROPERTY_DEFINITION('', '', #1208);
#1210=SHAPE_REPRESENTATION('', (#1207), #34);
#1211=SHAPE_DEFINITION_REPRESENTATION(#1209, #1210);
#1212=DIMENSIONAL_LOCATION('', '', #77, #1208);
#1213=PRECISION_QUALIFIER(4);
#1219=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1213
)) REPRESENTATION_ITEM(''));
#1220=SHAPE_DIMENSION_REPRESENTATION('', (#1219), #34);
#1221=PRECISION_QUALIFIER(4);
#1222=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0300000000000000), #15);
#1223=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0300000000000000), #15);
#1224=MEASURE_QUALIFICATION('', '', #1222, (#1221));
#1225=MEASURE_QUALIFICATION('', '', #1223, (#1221));
#1226=TOLERANCE_VALUE(#1223, #1222);
#1227=PLUS_MINUS_TOLERANCE(#1226, #1212);
#1228=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1212, #1220);
#1235=(CHARACTERIZED_OBJECT('', 'outer diameter to
shoulder') FEATURE_DEFINITION() INSTANCED_FEATURE() OUTER_ROUND() SHAPE_ASPECT('',
'outer diameter to shoulder', #43, .T.));
#1236=PROPERTY_DEFINITION('', '', #1235);
#1237=PRODUCT_DEFINITION_SHAPE('', '', #1235);
#1238=CARTESIAN_POINT('', (0.0, -3.235562E-32, 3.9400000000000000));
#1239=DIRECTION('', (0.0, 0.0, 1.0));
#1240=DIRECTION('', (0.0, -1.0, 0.0));
#1241=AXIS2_PLACEMENT_3D('orientation', #1238, #1239, #1240);
#1242=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0300000000000000);
#1243=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0300000000000000);
#1244=PRECISION_QUALIFIER(4);
#1250=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000), #15) QUALIFIED_REPRESENTATION_ITEM((#1242
, #1243, #1244)) REPRESENTATION_ITEM('diameter'));
#1251=SHAPE_REPRESENTATION('', (#814, #842), #34);
#1252=SHAPE_DEFINITION_REPRESENTATION(#1236, #1251);
#1253=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1241, #1250), #34);
#1254=SHAPE_DEFINITION_REPRESENTATION(#1236, #1253);
#1255=CARTESIAN_POINT('', (0.0, -0.1250000000000000, 3.9400000000000000));
#1256=DIRECTION('', (1.0, 0.0, 0.0));

```

```

#1257=DIRECTION('', (0.0,0.0,1.0));
#1258=AXIS2_PLACEMENT_3D('orientation', #1255, #1256, #1257);
#1259=FEATURE_COMPONENT_DEFINITION('', '');
#1260=PRODUCT_DEFINITION_SHAPE('', '', #1259);
#1261=VEE_PROFILE('vee profile', 'v-shape', #1260, .F.);
#1266=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#1271=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.0), #24)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#1276=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0), #15)REPRESENTATION_ITEM('profile radius'));
#1277=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1258, #1266, #1271, #1276), #34);
#1278=PROPERTY_DEFINITION('', '', #1261);
#1279=SHAPE_DEFINITION_REPRESENTATION(#1278, #1277);
#1280=SHAPE_ASPECT('', 'v-shape boundary occurrence', #1237, .T.);
#1281=SHAPE_DEFINING_RELATIONSHIP('vee boundary', 'profile usage', #1261, #1280);
#1282=CARTESIAN_POINT('', (0.0, -3.235562E-32, 3.9400000000000000));
#1283=DIRECTION('', (0.0,0.0,1.0));
#1284=DIRECTION('', (0.0,-1.0,0.0));
#1285=AXIS2_PLACEMENT_3D('', #1282, #1283, #1284);
#1286=SHAPE_ASPECT('', '', #43, .F.);
#1287=PROPERTY_DEFINITION('', '', #1286);
#1288=SHAPE_REPRESENTATION('', (#1285), #34);
#1289=SHAPE_DEFINITION_REPRESENTATION(#1287, #1288);
#1290=DIMENSIONAL_LOCATION('', '', #1286, #1032);
#1291=PRECISION_QUALIFIER(4);
#1297=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(3.6900000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1291))REPRESENTATION_ITEM(''));
#1298=SHAPE_DIMENSION_REPRESENTATION('', (#1297), #34);
#1299=PRECISION_QUALIFIER(4);
#1300=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000), #15);
#1301=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0100000000000000), #15);
#1302=MEASURE_QUALIFICATION('', '', #1300, (#1299));
#1303=MEASURE_QUALIFICATION('', '', #1301, (#1299));
#1304=TOLERANCE_VALUE(#1301, #1300);
#1305=PLUS_MINUS_TOLERANCE(#1304, #1290);
#1306=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1290, #1298);
#1313=(CHARACTERIZED_OBJECT('', 'outer diameter')FEATURE_DEFINITION()INSTANCED_FEATURE()OUTER_ROUND()SHAPE_ASPECT('', 'outer diameter', #43, .T.));
#1314=PROPERTY_DEFINITION('', '', #1313);
#1315=PRODUCT_DEFINITION_SHAPE('', '', #1313);
#1316=CARTESIAN_POINT('', (0.0, -3.235562E-32, 3.7999999999999999));
#1317=DIRECTION('', (0.0,0.0,1.0));
#1318=DIRECTION('', (0.0,-1.0,0.0));
#1319=AXIS2_PLACEMENT_3D('orientation', #1316, #1317, #1318);
#1320=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#1321=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#1322=PRECISION_QUALIFIER(4);

```



```

#1328=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.3720000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1320
,#1321,#1322))REPRESENTATION_ITEM('diameter'));
#1329=STANDARD_UNCERTAINTY('upper limit','plus',0.0100000000000000);
#1330=STANDARD_UNCERTAINTY('lower limit','minus',-0.0100000000000000);
#1331=PRECISION_QUALIFIER(4);
#1337=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1400000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1329
,#1330,#1331))REPRESENTATION_ITEM('length'));
#1338=SHAPE_REPRESENTATION('',(#907,#924),#34);
#1339=SHAPE_DEFINITION_REPRESENTATION(#1314,#1338);
#1340=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1319,#1328,#1337),#34);
#1341=SHAPE_DEFINITION_REPRESENTATION(#1314,#1340);
#1342=FEATURE_COMPONENT_DEFINITION('','');
#1343=PRODUCT_DEFINITION_SHAPE('','',#1342);
#1344=TAPER('','diameter taper',#1343,.F.);
#1345=STANDARD_UNCERTAINTY('upper limit','plus',0.0300000000000000);
#1346=STANDARD_UNCERTAINTY('lower limit','minus',-0.0300000000000000);
#1347=PRECISION_QUALIFIER(4);
#1353=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1345
,#1346,#1347))REPRESENTATION_ITEM('final diameter'));
#1354=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1353),#34);
#1355=PROPERTY_DEFINITION('','',#1344);
#1356=SHAPE_DEFINITION_REPRESENTATION(#1355,#1354);
#1357=SHAPE_ASPECT('','reduced size occurrence',#1315,.T.);
#1358=FEATURE_COMPONENT_RELATIONSHIP('reduced size','taper
usage',#1344,#1357);

#1365=(CHARACTERIZED_OBJECT('','')FEATURE_DEFINITION()INSTANCED_FEATURE()ROUN
D_HOLE()SHAPE_ASPECT('','',#43,.T.));
#1366=PROPERTY_DEFINITION('','',#1365);
#1367=PRODUCT_DEFINITION_SHAPE('','',#1365);
#1368=CARTESIAN_POINT('',(0.1860000000000000,1.138922E-17,2.5800000000000000));
#1369=DIRECTION('',(-1.0,0.0,0.0));
#1370=DIRECTION('',(0.0,-1.0,0.0));
#1371=AXIS2_PLACEMENT_3D('orientation',#1368,#1369,#1370);
#1372=SHAPE_REPRESENTATION('',(#950,#968),#34);
#1373=SHAPE_DEFINITION_REPRESENTATION(#1366,#1372);
#1374=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371),#34);
#1375=SHAPE_DEFINITION_REPRESENTATION(#1366,#1374);
#1376=FEATURE_COMPONENT_DEFINITION('','');
#1377=PRODUCT_DEFINITION_SHAPE('','',#1376);
#1378=CIRCULAR_CLOSED_PROFILE('circular profile','',#1377,.F.);
#1379=STANDARD_UNCERTAINTY('upper limit','plus',0.0100000000000000);
#1380=STANDARD_UNCERTAINTY('lower limit','minus',-0.0100000000000000);
#1381=PRECISION_QUALIFIER(4);
#1387=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1250000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1379
,#1380,#1381))REPRESENTATION_ITEM('diameter'));
#1388=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#1387),#34);
#1389=PROPERTY_DEFINITION('','',#1378);
#1390=SHAPE_DEFINITION_REPRESENTATION(#1389,#1388);

```

```

#1391=FEATURE_COMPONENT_DEFINITION('', '');
#1392=PRODUCT_DEFINITION_SHAPE('', '#1391');
#1393=PATH_FEATURE_COMPONENT('linear path', 'linear', #1392, .F.);
#1398=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.3720000000000000), #15)REPRESENTATION_ITEM('distance'));
#1399=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #1398), #34);
#1400=PROPERTY_DEFINITION('', '#1393');
#1401=SHAPE_DEFINITION_REPRESENTATION(#1400, #1399);
#1402=DIRECTION_SHAPE_REPRESENTATION('', (#1369), #34);
#1403=PROPERTY_DEFINITION_REPRESENTATION(#1400, #1402);
#1404=SHAPE_ASPECT('', 'diameter occurrence', #1367, .T.);
#1405=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #1378, #1404);
#1406=SHAPE_ASPECT('', 'hole depth occurrence', #1367, .T.);
#1407=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #1393, #1406);
#1408=FEATURE_COMPONENT_DEFINITION('', '');
#1409=PRODUCT_DEFINITION_SHAPE('', '#1408');
#1410=HOLE_BOTTOM('bottom condition', 'through', #1409, .F.);
#1411=SHAPE_ASPECT('', 'bottom condition occurrence', #1367, .T.);
#1412=FEATURE_COMPONENT_RELATIONSHIP('', 'hole bottom usage', #1410, #1411);
#1413=CARTESIAN_POINT('', (0.1860000000000000, 1.387779E-17, 2.5800000000000000));
#1414=DIRECTION('', (-1.0, 0.0, 0.0));
#1415=AXIS1_PLACEMENT('', #1413, #1414);
#1416=SHAPE_ASPECT('', '#43, .F.);
#1417=PROPERTY_DEFINITION('', '#1416);
#1418=SHAPE_REPRESENTATION('', (#1415), #34);
#1419=SHAPE_DEFINITION_REPRESENTATION(#1417, #1418);
#1420=DIMENSIONAL_LOCATION('', '#1032, #1416);
#1421=PRECISION_QUALIFIER(4);
#1427=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.3300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1421
))REPRESENTATION_ITEM(''));
#1428=SHAPE_DIMENSION_REPRESENTATION('', (#1427), #34);
#1429=PRECISION_QUALIFIER(4);
#1430=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0300000000000000), #15);
#1431=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0300000000000000), #15);
#1432=MEASURE_QUALIFICATION('', '#1430, (#1429));
#1433=MEASURE_QUALIFICATION('', '#1431, (#1429));
#1434=TOLERANCE_VALUE(#1431, #1430);
#1435=PLUS_MINUS_TOLERANCE(#1434, #1420);
#1436=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#1420, #1428);
#1437=CHAMFER('', '#43, .T.);
#1438=PROPERTY_DEFINITION('', '#1437);
#1439=FACE_SHAPE_REPRESENTATION('chamfer face', (#866), #34);
#1440=PROPERTY_DEFINITION_REPRESENTATION(#1438, #1439);
#1441=CHAMFER_OFFSET('', 'first offset', #43, .F.);
#1442=CHAMFER_OFFSET('', 'second offset', #43, .F.);
#1443=FEATURE_COMPONENT_RELATIONSHIP('', '#1437, #1441);
#1444=FEATURE_COMPONENT_RELATIONSHIP('', '#1437, #1442);
#1445=PROPERTY_DEFINITION('', '#1441);
#1446=PROPERTY_DEFINITION('', '#1442);
#1447=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0300000000000000);
#1448=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0300000000000000);

```

```

#1449=PRECISION_QUALIFIER(4);
#1455=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1447
,#1448,#1449))REPRESENTATION_ITEM('offset amount'));
#1456=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1455),#34);
#1457=SHAPE_DEFINITION_REPRESENTATION(#1445,#1456);
#1458=FACE_SHAPE_REPRESENTATION('first chamfer face',( #814),#34);
#1459=PROPERTY_DEFINITION_REPRESENTATION(#1445,#1458);
#1460=STANDARD_UNCERTAINTY('upper limit','plus',0.5000000000000000);
#1461=STANDARD_UNCERTAINTY('lower limit','minus',-0.5000000000000000);
#1462=PRECISION_QUALIFIER(4);
#1468=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(15.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED_REPRESENTATION_ITEM((#1460,#1
461,#1462))REPRESENTATION_ITEM('offset angle'));
#1469=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1468),#34);
#1470=SHAPE_DEFINITION_REPRESENTATION(#1446,#1469);
#1471=FACE_SHAPE_REPRESENTATION('second chamfer face',( #138),#34);
#1472=PROPERTY_DEFINITION_REPRESENTATION(#1446,#1471);
#1473=CHAMFER('','',#43,.T.);
#1474=PROPERTY_DEFINITION('','',#1473);
#1475=FACE_SHAPE_REPRESENTATION('chamfer face',( #882),#34);
#1476=PROPERTY_DEFINITION_REPRESENTATION(#1474,#1475);
#1477=CHAMFER_OFFSET('','first offset',#43,.F.);
#1478=CHAMFER_OFFSET('','second offset',#43,.F.);
#1479=FEATURE_COMPONENT_RELATIONSHIP('','',#1473,#1477);
#1480=FEATURE_COMPONENT_RELATIONSHIP('','',#1473,#1478);
#1481=PROPERTY_DEFINITION('','',#1477);
#1482=PROPERTY_DEFINITION('','',#1478);
#1483=STANDARD_UNCERTAINTY('upper limit','plus',0.0300000000000000);
#1484=STANDARD_UNCERTAINTY('lower limit','minus',-0.0300000000000000);
#1485=PRECISION_QUALIFIER(4);
#1491=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('offset amount'));
#1492=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1491),#34);
#1493=SHAPE_DEFINITION_REPRESENTATION(#1481,#1492);
#1494=FACE_SHAPE_REPRESENTATION('first chamfer face',( #842),#34);
#1495=PROPERTY_DEFINITION_REPRESENTATION(#1481,#1494);
#1496=STANDARD_UNCERTAINTY('upper limit','plus',0.5000000000000000);
#1497=STANDARD_UNCERTAINTY('lower limit','minus',-0.5000000000000000);
#1498=PRECISION_QUALIFIER(4);
#1504=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(15.
0),#24)PLANE_ANGLE_MEASURE_WITH_UNIT()QUALIFIED_REPRESENTATION_ITEM((#1496,#1
497,#1498))REPRESENTATION_ITEM('offset angle'));
#1505=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1504),#34);
#1506=SHAPE_DEFINITION_REPRESENTATION(#1482,#1505);
#1507=FACE_SHAPE_REPRESENTATION('second chamfer face',( #138),#34);
#1508=PROPERTY_DEFINITION_REPRESENTATION(#1482,#1507);
#1509=SHAPE_ASPECT_RELATIONSHIP('','',#1473,#1437);
#1510=DESCRIPTIVE_REPRESENTATION_ITEM('note text','BREAK ALL SHARP EDGES
APPOX .015.');
```

```

#1511=REPRESENTATION('note text',( #1510),#34);
#1512=PROPERTY_DEFINITION('part property','part note',#42);

```

ISO 10303-223:2008 (E)

```
#1513=PROPERTY_DEFINITION('part property','part note',#42);
#1514=PROPERTY_DEFINITION_REPRESENTATION(#1513,#1511);
#1515=PROPERTY_DEFINITION_RELATIONSHIP('', '#1512', #1513);
#1516=DESCRIPTIVE_REPRESENTATION_ITEM('note text','HEAT TREAT PER MIL-H-6875
TO 150,000-160,000 PSI. ');
#1517=REPRESENTATION('note text', (#1516), #34);
#1518=PROPERTY_DEFINITION('part property','part note',#42);
#1519=PROPERTY_DEFINITION('part property','part note',#42);
#1520=PROPERTY_DEFINITION_REPRESENTATION(#1519,#1517);
#1521=PROPERTY_DEFINITION_RELATIONSHIP('', '#1518', #1519);
#1522=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#1523=PRODUCT_DEFINITION('3/8 OD X 4 5/16 LG', '4130 STEEL COND D & 4 PER MIL-
S-6758', #40, #41);
#1524=MATERIAL_DESIGNATION('4130 STEEL COND D & 4 PER MIL-S-6758', (#1523));
#1525=MAKE_FROM_USAGE_OPTION('1', '#1523', 'product material', #42, #1523, 1, 'primary
material', #1522);
#1526=PRODUCT_DEFINITION_SHAPE('base shape', 'base shape for part', #1523);

#1528=VERSIONED_ACTION_REQUEST('', '#1528');
#1529=ACTION_DIRECTIVE('', '#1528');
#1530=ACTION_METHOD('', '#1530');
#1531=DIRECTED_ACTION('project order', 'order for project', #1530, #1529);

#1533=APPROVAL_STATUS('approved');
#1534=APPROVAL(#1533, 'product version approval');

#1536=APPROVAL(#1533, 'project order approval');

#1538=CALENDAR_DATE(1997, 1, 8);
#1539=APPROVAL_DATE_TIME(#1538, #1534);
#1540=APPROVAL_DATE_TIME(#1538, #1536);
#1541=PERSON('1', 'Operator', 'RPTS', $, $, $);
#1542=ORGANIZATION($, 'RPTS', 'Producer');
#1543=PERSON_AND_ORGANIZATION(#1541, #1542);
#1544=APPROVAL_ROLE('approval product version');
#1545=APPROVAL_PERSON_ORGANIZATION(#1543, #1534, #1544);
#1546=APPROVAL_PERSON_ORGANIZATION(#1543, #1536, #1544);
#1547=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#1548=SECURITY_CLASSIFICATION('classification', 'classify product
version', #1547);

#1551=MEASURE_REPRESENTATION_ITEM('quantity', COUNT_MEASURE(1.0), #35);
#1552=REPRESENTATION('quantity', (#1551), #34);

#1555=ORGANIZATION_ROLE('manufacturer');

#1557=ORGANIZATION_ROLE('owner');

#1559=PERSON_AND_ORGANIZATION_ROLE('manufacturer');

#1561=PERSON_AND_ORGANIZATION_ROLE('owner');
```

```

/* ----- casting_envelope_requirement -----*/

#1562=CASTING_ENVELOPE_REQUIREMENT('', '#42');
#1563=PROPERTY_DEFINITION_REPRESENTATION(#1562, #1570);

#1570=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1572, #1574, #1576), #34);
#1572=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('part height'));
#1574=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('part width'));
#1576=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('part length'));

/* ----- request_for_clarification -----*/

#1580=VERSIONED_ACTION_REQUEST('', 'request for clarification', 'request
description');
#1581=APPLIED_DATE_ASSIGNMENT(#1582, #1583, (#1580));
#1582=DATE(2006);
#1583=DATE_ROLE('request date');
#1584=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#1586, #1589, (#1580));
#1586=PERSON_AND_ORGANIZATION(#1588, #1587);
#1587=ORGANIZATION('NGIT', 'Northrop Grumman', '');
#1588=PERSON('1', 'Slovensky', 'Len', ('W'), ('Dr'), (''));
#1589=PERSON_AND_ORGANIZATION_ROLE('customer');
#1590=APPLIED_ACTION_REQUEST_ASSIGNMENT(#1580, (#42));

/* ----- casting_insert -----*/

#2000=(CASTING_FEATURE_DEFINITION()
  CASTING_INSERT()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '#30143, .T.));

#2012=MATERIAL_DESIGNATION('', (#2000, #2050));

#2013=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('draft'));
#2014=DESCRIPTIVE_REPRESENTATION_ITEM('insert name', 'CASTING INSERT 123456');

```

ISO 10303-223:2008 (E)

```

#2015=DESCRIPTIVE_REPRESENTATION_ITEM('positioning method','POSITONING METHOD
123456');
#2016=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#2013,#2014,#2015),#34);
#2018=PROPERTY_DEFINITION('', '#2000);
#2020=PROPERTY_DEFINITION_REPRESENTATION(#2018,#2016);

/* ----- mould_box -----*/

#2030=(CASTING_FEATURE_DEFINITION()
CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '#2000);
MOULD_BOX()
SAND_CAST_DESIGN_FEATURE()
SHAPE_ASPECT('', '#43, .T.));
#2032=PRODUCT_DEFINITION_SHAPE('', '#2030);

#2036=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371),#34);
#2038=PROPERTY_DEFINITION('', '#2030);
#2040=PROPERTY_DEFINITION_REPRESENTATION(#2038,#2036);

#2044=SHAPE_ASPECT('', 'mould box occurrence',#2032, .T.);
#3346=FEATURE_COMPONENT_RELATIONSHIP('', 'mould box dimension
usage', #3202, #2044);

/* ----- chill -----*/

#2050=(CASTING_FEATURE_DEFINITION()
CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '#2000);
CHILL()
SAND_CAST_DESIGN_FEATURE()
SHAPE_ASPECT('', '#43, .T.));
#2051=PRODUCT_DEFINITION_SHAPE('', '#2050);
#2053=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('draft'));
#2054=DESCRIPTIVE_REPRESENTATION_ITEM('internal or external chill','true');
#2056=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#2053,#2054),#34);
#2058=PROPERTY_DEFINITION('', '#2050);
#2060=PROPERTY_DEFINITION_REPRESENTATION(#2058,#2056);

#2064=SHAPE_ASPECT('', 'chill area occurrence',#2051, .T.);
#2066=SHAPE_DEFINING_RELATIONSHIP('', 'chill area shape usage',#561,#2064);
#2070=SHAPE_DEFINING_RELATIONSHIP('', 'chill area shape usage',#561,#2064);

/* ----- loose_piece -----*/

#3000=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()

CHARACTERIZED_OBJECT('', '#2000);LOOSE_PIECE()SAND_CAST_DESIGN_FEATURE()SHAPE_ASPEC
T('', '#43, .T.));
#3002=PRODUCT_DEFINITION_SHAPE('', '#3000);

```

```

#3012=MATERIAL_DESIGNATION('wood',(#3000));
#3014=SHAPE_ASPECT('', 'loose piece shape occurrence',#3002,.T.);
#3016=SHAPE_DEFINING_RELATIONSHIP('', 'loose piece shape usage',#561,#3014);

#3056=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371),#34);
#3058=PROPERTY_DEFINITION('', '',#3000);
#3060=PROPERTY_DEFINITION_REPRESENTATION(#3058,#3056);

/* ----- chaplet_pad -----*/

#3100=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()CHAPLET_PAD()
CHARACTERIZED_OBJECT('', '')SAND_CAST_DESIGN_FEATURE()SHAPE_ASPECT('', '',#43,.
T.));
#3102=PRODUCT_DEFINITION_SHAPE('', '',#3100);

#3156=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371),#34);
#3158=PROPERTY_DEFINITION('', '',#3100);
#3160=PROPERTY_DEFINITION_REPRESENTATION(#3158,#3156);

/* ----- casting_feature_size -----*/

#3200=FEATURE_COMPONENT_DEFINITION('', '');
#3201=PRODUCT_DEFINITION_SHAPE('', '',#3200);

#3202=CASTING_FEATURE_SIZE('', '',#3201,.F.);
#3203=SHAPE_ASPECT('', 'chaplet pad dimension occurrence',#3102,.T.);
#3204=FEATURE_COMPONENT_RELATIONSHIP('', 'chaplet          pad          dimension
usage', #3202,#3203);

#3206=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#3212,#3214,#3216,#3218,#3220)
,#34);
#3208=PROPERTY_DEFINITION('', '',#3202);
#3210=PROPERTY_DEFINITION_REPRESENTATION(#3208,#3206);
#3212=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('item height'));
#3214=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('item width'));
#3216=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('item length'));
#3218=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('item diameter'));
#3220=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000),#15)QUALIFIED_REPRESENTATION_ITEM((#1483
,#1484,#1485))REPRESENTATION_ITEM('item draft'));

#3230=FEATURE_COMPONENT_DEFINITION('', '');
#3231=PRODUCT_DEFINITION_SHAPE('', '',#3230);

```

ISO 10303-223:2008 (E)

```

#3232=CASTING_FEATURE_SIZE('', '#3231', .F.);
#3238=PROPERTY_DEFINITION('', '#3232');
#3240=PROPERTY_DEFINITION_REPRESENTATION(#3238, #3206);

/* ----- chaplet -----*/

#3300=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()CHAPLET()
CHARACTERIZED_OBJECT('', 'SAND_CAST_DESIGN_FEATURE()SHAPE_ASPECT('', '#43, .
T.));
#3302=PRODUCT_DEFINITION_SHAPE('', '#3300);

#3304=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3309), #34);
#3306=PROPERTY_DEFINITION('', '#3300);
#3308=PROPERTY_DEFINITION_REPRESENTATION(#3306, #3304);
#3309=DESCRIPTIVE_REPRESENTATION_ITEM('chaplet type', 'metal');

#3310=SHAPE_ASPECT('', 'chaplet reference occurrence', #3302, .T.);
#3312=FEATURE_COMPONENT_RELATIONSHIP('', 'chaplet dimension
usage', #3202, #3310);
#3316=SHAPE_DEFINING_RELATIONSHIP('', 'chaplet pad reference
usage', #3100, #3310);

/* ----- pin_center -----*/

#3330=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', 'PIN_CENTER()SAND_CAST_DESIGN_FEATURE()SHAPE_ASPECT
('', '#43, .T.));
#3332=PRODUCT_DEFINITION_SHAPE('', '#3330);

#3334=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3340), #34);
#3336=PROPERTY_DEFINITION('', '#3330);
#3338=PROPERTY_DEFINITION_REPRESENTATION(#3336, #3334);
#3340=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485)REPRESENTATION_ITEM('dimension'));

/* ----- flask -----*/

#3350=(CASTING_FEATURE_DEFINITION()
CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', 'FLASK()
SAND_CAST_DESIGN_FEATURE()
SHAPE_ASPECT('', '#43, .T.));

#3352=PRODUCT_DEFINITION_SHAPE('', '#3350);

#3354=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3359, #3360, #3361, #3362,
#3363), #34);

```



```

#3356=PROPERTY_DEFINITION('', '#3350');
#3358=PROPERTY_DEFINITION_REPRESENTATION(#3356, #3354);
#3359=DESCRIPTIVE_REPRESENTATION_ITEM('flask type', 'metal');
#3360=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('cope height'));
#3361=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('drag height'));
#3362=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('flask length'));
#3363=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('flask width'));

#3374=SHAPE_ASPECT('', 'flask reference occurrence', #3352, .T.);
#3376=SHAPE_DEFINING_RELATIONSHIP('', 'pin                center                reference
usage', #3330, #3374);
#3378=SHAPE_DEFINING_RELATIONSHIP('', 'sand                mould                reference
usage', #3400, #3374);
/* ----- mould_lock -----*/

#3380=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  MOULD_LOCK()
  SAND_CAST_DESIGN_FEATURE()
  SHAPE_ASPECT('', '#43, .T.));
#3382=PRODUCT_DEFINITION_SHAPE('', '#3380);

#3384=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3393), #34);
#3386=PROPERTY_DEFINITION('', '#3380);
#3388=PROPERTY_DEFINITION_REPRESENTATION(#3386, #3384);

#3390=SHAPE_ASPECT('', 'mould lock occurrence', #3382, .T.);
#3392=FEATURE_COMPONENT_RELATIONSHIP('', 'mould                lock                dimension
usage', #3202, #3390);
#3393=DESCRIPTIVE_REPRESENTATION_ITEM('mould lock description', 'metal');

/* ----- sand mould -----*/

#3400=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  SAND_CAST_DESIGN_FEATURE(
  ) SAND_MOULD()
  SHAPE_ASPECT('', '#43, .T.));

#3402=PRODUCT_DEFINITION_SHAPE('', '#3400);

#3404=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3409), #34);

```

ISO 10303-223:2008 (E)

```

#3406=PROPERTY_DEFINITION('', '#3400');
#3408=PROPERTY_DEFINITION_REPRESENTATION(#3406, #3404);

#3409=DESCRIPTIVE_REPRESENTATION_ITEM('sand      mould      type', 'grean      sand
material');

#3410=SHAPE_ASPECT('', 'sand mould occurrence', #3402, .T.);
#3412=SHAPE_DEFINING_RELATIONSHIP('', 'chaplet reference usage', #3300, #3410);
#3416=SHAPE_DEFINING_RELATIONSHIP('', 'chill reference usage', #2050, #3410);

/* ----- moulding_equipment -----*/

#3430=MOULDING_EQUIPMENT('', '#3434, #30577, #19501', #3432);

#3432=ACTION_RESOURCE_TYPE('');
#3434=ACTION_METHOD('', '#3438, #3400, #3500');
#3436=APPLIED_ACTION_METHOD_ASSIGNMENT(#3434, #3438, (#3400, #3500));
#3438=ACTION_METHOD_ROLE('', '#3438');

#3440=ORGANIZATION($, 'RPTS', 'equipment company name');
#3441=ORGANIZATION_ROLE('owner');
#3442=APPLIED_ORGANIZATION_ASSIGNMENT(#3440, #3441, (#3430, #3630, #5180, #9000, #9
100, #9200, #9300, #9400, #9500, #9700, #9800));

#3443=RESOURCE_PROPERTY('', '#3430');
#3444=RESOURCE_PROPERTY_REPRESENTATION('', '#3443, #9035);

#3446=MACHINE_ELEMENT_RELATIONSHIP('', '#11000, #3430);
#3447=RESOURCE_PROPERTY('', '#3430);
#3448=RESOURCE_PROPERTY_REPRESENTATION('location', '#3447, #5087);

/* ----- pattern_plate -----*/

#3500=(CASTING_FEATURE_DEFINITION()
      CASTING_INSTANCED_FEATURE()
      CHARACTERIZED_OBJECT('', '#3500')
      PATTERN_PLATE()
      SAND_CAST_DESIGN_FEATURE()
      SHAPE_ASPECT('', '#43, .T.));
#3502=PRODUCT_DEFINITION_SHAPE('', '#3500);

#3504=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#3506=PROPERTY_DEFINITION('', '#3500);
#3508=PROPERTY_DEFINITION_REPRESENTATION(#3506, #3504);

#3510=MATERIAL_DESIGNATION('wood', (#3500));

#3590=SHAPE_ASPECT('', 'pattern plate occurrence', #3502, .T.);
#3592=FEATURE_COMPONENT_RELATIONSHIP('', 'pattern plate usage', #3202, #3590);
#3604=SHAPE_DEFINING_RELATIONSHIP('', 'pin      center      reference
usage', #3330, #3590);

```

```

#3607=SHAPE_DEFINING_RELATIONSHIP('', 'mould          lock          reference
usage', #3380, #3590);
#3609=SHAPE_DEFINING_RELATIONSHIP('', 'flask reference usage', #3350, #3590);
#3610=SHAPE_DEFINING_RELATIONSHIP('', 'gating          system          reference
usage', #4000, #3590);

#3611=PROPERTY_DEFINITION_REPRESENTATION(#3506, #3612);
#3612=PARTING_SURFACE('parting line', (#600, #3613, #3614, #3616), #34);
#3613=DESCRIPTIVE_REPRESENTATION_ITEM('cope or drag', 'cope');
#3614=DESCRIPTIVE_REPRESENTATION_ITEM('male or female', 'male');
#3616=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('offset distance'));
#3617=REPRESENTATION_RELATIONSHIP('', '', #3612, #79);

#3624=PROPERTY_DEFINITION_REPRESENTATION(#3506, #3626);
#3626=LOCATION_SHAPE_REPRESENTATION('location', (#595), #34);

/* ----- dryer_equipment -----*/

#3630=(ACTION_RESOURCE('', '', (#3634), #3632)
  CASTING_EQUIPMENT()
  CORE_EQUIPMENT()
  DRYER()
  MACHINE());

#3632=ACTION_RESOURCE_TYPE('');
#3634=ACTION_METHOD('', '', '', '');
#3636=APPLIED_ACTION_METHOD_ASSIGNMENT(#3634, #3638, (#3500));
#3638=ACTION_METHOD_ROLE('', '');

#3643=RESOURCE_PROPERTY('', '', #3630);
#3644=RESOURCE_PROPERTY_REPRESENTATION('', '', #3643, #9035);

#3646=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #3630);
#3647=RESOURCE_PROPERTY('', '', #3630);
#3648=RESOURCE_PROPERTY_REPRESENTATION('location', '', #3647, #5087);

#3651=RESOURCE_PROPERTY_REPRESENTATION('', '', #3643, #3652);
#3652=REPRESENTATION('equipment type', (#3653), #34);
#3653=DESCRIPTIVE_REPRESENTATION_ITEM('material type', ' ');

/* ----- core -----*/

#3700=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  CORE()
  SAND_CAST_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.));

```

ISO 10303-223:2008 (E)

```

#3702=PRODUCT_DEFINITION_SHAPE('', '#3700');

#3704=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3709), #34);
#3706=PROPERTY_DEFINITION('', '#3700');
#3708=PROPERTY_DEFINITION_REPRESENTATION(#3706, #3704);
#3709=DESCRIPTIVE_REPRESENTATION_ITEM('core type', 'ram up core');

#3710=MATERIAL_DESIGNATION('plaster', (#3700));

#3714=PROPERTY_DEFINITION_REPRESENTATION(#3706, #3716);
#3716=LOCATION_SHAPE_REPRESENTATION('vent location', (#595), #34);

#3718=SHAPE_ASPECT('', 'core occurrence', #3702, .T.);
#3720=SHAPE_DEFINING_RELATIONSHIP('', 'core placement reference
usage', #3400, #3718);

#3792=FEATURE_COMPONENT_RELATIONSHIP('', 'gas vent size usage', #3202, #3718);

/* ----- core_print -----*/

#3800=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', 'CORE_PRINT()SAND_CAST_DESIGN_FEATURE()SHAPE_ASPECT
('', '#43, .T.));
#3802=PRODUCT_DEFINITION_SHAPE('', '#3800);

#3804=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #3809), #34);
#3806=PROPERTY_DEFINITION('', '#3800);
#3808=PROPERTY_DEFINITION_REPRESENTATION(#3806, #3804);

#3809=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('clearance'));

#3811=PROPERTY_DEFINITION_REPRESENTATION(#3806, #3812);
#3812=DRAFTED_SURFACE('draft', (#600, #3813), #34);
#3813=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
REPRESENTATION_ITEM('draft'));
#3815=REPRESENTATION_RELATIONSHIP('', '#3812, #5139);

#3894=SHAPE_ASPECT('', 'core print occurrence', #3802, .T.);
#3896=SHAPE_DEFINING_RELATIONSHIP('', 'core print shape usage', #561, #3894);
#3892=FEATURE_COMPONENT_RELATIONSHIP('', 'core print size usage', #3202, #3894);

/* ----- gating_system -----*/

#4000=(CASTING_FEATURE_DEFINITION()
CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '

```

```

GATING_DESIGN_FEATURE()
GATING_SYSTEM()
SHAPE_ASPECT('', '#33008, .T.);

#4002=PRODUCT_DEFINITION_SHAPE('', '#4000);

#4004=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#4006=PROPERTY_DEFINITION('', '#4000);
#4008=PROPERTY_DEFINITION_REPRESENTATION(#4006, #4004);

#4026=SHAPE_ASPECT('', 'gating system occurrence', #4002, .T.);
#4028=SHAPE_DEFINING_RELATIONSHIP('', 'gating system components
usage', #4200, #4026);

/* ----- well -----*/

#4100=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', 'GATING_DESIGN_FEATURE()SHAPE_ASPECT('', '#33008, .
T.)WELL());
#4102=PRODUCT_DEFINITION_SHAPE('', '#4100);

#4104=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#4106=PROPERTY_DEFINITION('', '#4100);
#4108=PROPERTY_DEFINITION_REPRESENTATION(#4106, #4104);

#4114=SHAPE_ASPECT('', 'well occurrence', #4102, .T.);
#4116=FEATURE_COMPONENT_RELATIONSHIP('', 'well size usage', #3202, #4114);

#4118=PROPERTY_DEFINITION_REPRESENTATION(#4106, #3612);

/* ----- sprue -----*/

#4200=(CASTING_FEATURE_DEFINITION()
CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '
GATING_DESIGN_FEATURE()
SHAPE_ASPECT('', '#43, .T.)
SPRUE());
#4202=PRODUCT_DEFINITION_SHAPE('', '#4200);

#4204=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4209), #34);
#4206=PROPERTY_DEFINITION('', '#4200);
#4208=PROPERTY_DEFINITION_REPRESENTATION(#4206, #4204);
#4209=DESCRIPTIVE_REPRESENTATION_ITEM('sprue core', 'good stuff');
#4214=SHAPE_ASPECT('', 'sprue occurrence', #4202, .T.);
#4216=FEATURE_COMPONENT_RELATIONSHIP('', 'sprue size usage', #3202, #4214);
#4217=FEATURE_COMPONENT_RELATIONSHIP('', 'sprue pouring cup size
usage', #3202, #4214);

#4224=SHAPE_DEFINING_RELATIONSHIP('', 'filter reference usage', #4300, #4214);
#4228=SHAPE_DEFINING_RELATIONSHIP('', 'well reference usage', #4100, #4214);

```

```

/* ----- filter -----*/

#4300=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  FILTER()
  GATING_DESIGN_FEATURE()
  SHAPE_ASPECT('','',#43,.T.));
#4302=PRODUCT_DEFINITION_SHAPE('','',#4300);

#4304=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(1371,#4309),#34);
#4306=PROPERTY_DEFINITION('','',#4300);
#4308=PROPERTY_DEFINITION_REPRESENTATION(#4306,#4304);
#4309=DESCRIPTIVE_REPRESENTATION_ITEM('filter type','metal screen');

#4314=SHAPE_ASPECT('','filer occurrence',#4302,.T.);
#4316=FEATURE_COMPONENT_RELATIONSHIP('','filter size usage',#3202,#4314);

/* ----- riser_contact -----*/

#4400=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  GATING_DESIGN_FEATURE()
  RISER_CONTACT()
  SHAPE_ASPECT('','',#43,.T.));
#4402=PRODUCT_DEFINITION_SHAPE('','',#4400);

#4404=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(1371,#4409),#34);
#4406=PROPERTY_DEFINITION('','',#4400);
#4408=PROPERTY_DEFINITION_REPRESENTATION(#4406,#4404);
#4409=DESCRIPTIVE_REPRESENTATION_ITEM('cope or drag','cope');

#4414=SHAPE_ASPECT('','riser contact occurrence',#4402,.T.);
#4416=SHAPE_DEFINING_RELATIONSHIP('','riser contact shape usage',#561,#4414);

/* ----- riser -----*/

#4500=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('','')GATING_DESIGN_FEATURE()RISER()SHAPE_ASPECT('','',#
43,.T.));
#4502=PRODUCT_DEFINITION_SHAPE('','',#4500);

#4504=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(1371,#4509,#4510),#34);
#4506=PROPERTY_DEFINITION('','',#4500);
#4508=PROPERTY_DEFINITION_REPRESENTATION(#4506,#4504);
#4509=DESCRIPTIVE_REPRESENTATION_ITEM('riser type','open');
#4510=DESCRIPTIVE_REPRESENTATION_ITEM('riser sleeve','insulating');
#4514=SHAPE_ASPECT('','riser occurrence',#4502,.T.);
#4516=FEATURE_COMPONENT_RELATIONSHIP('','riser size usage',#3202,#4514);

```

```

#4520=FEATURE_COMPONENT_RELATIONSHIP('', 'riser neck size usage', #3202, #4514);

#4528=SHAPE_DEFINING_RELATIONSHIP('', 'riser          contact          reference
usage', #4400, #4514);

#4538=SHAPE_DEFINING_RELATIONSHIP('', 'well reference usage', #4100, #4514);

/* ----- feeder -----*/

#4600=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '')FEEDER()GATING_DESIGN_FEATURE()SHAPE_ASPECT('', '',
#43, .T.));
#4602=PRODUCT_DEFINITION_SHAPE('', '', #4600);

#4604=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4610, #4611), #34);
#4606=PROPERTY_DEFINITION('', '', #4600);
#4608=PROPERTY_DEFINITION_REPRESENTATION(#4606, #4604);
#4610=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('edge round'));
#4611=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.1300000000000000), #15)QUALIFIED_REPRESENTATION_ITEM((#1483
, #1484, #1485))REPRESENTATION_ITEM('fillet'));

#4614=SHAPE_ASPECT('', 'feeder occurrence', #4602, .T.);
#4616=FEATURE_COMPONENT_RELATIONSHIP('', 'feeder size usage', #3202, #4614);

#4628=SHAPE_DEFINING_RELATIONSHIP('', 'filter reference usage', #4300, #4614);

#4638=SHAPE_DEFINING_RELATIONSHIP('', 'riser reference usage', #4500, #4614);

/* ----- runner -----*/

#4700=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '')GATING_DESIGN_FEATURE()RUNNER()SHAPE_ASPECT('', '',
#43, .T.));
#4702=PRODUCT_DEFINITION_SHAPE('', '', #4700);

#4704=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4709), #34);
#4706=PROPERTY_DEFINITION('', '', #4700);
#4707=PROPERTY_DEFINITION_REPRESENTATION(#4706, #3716);
#4708=PROPERTY_DEFINITION_REPRESENTATION(#4706, #4704);
#4709=DESCRIPTIVE_REPRESENTATION_ITEM('cope or drag', 'cope');

#4710=SHAPE_ASPECT('', 'runner occurrence', #4702, .T.);
#4712=FEATURE_COMPONENT_RELATIONSHIP('', 'runner size usage', #3202, #4710);

#4715=FEATURE_COMPONENT_RELATIONSHIP('', 'runner          dimensions          size
usage', #3202, #4710);

#4718=SHAPE_DEFINING_RELATIONSHIP('', 'feeder reference usage', #4600, #4710);

```

ISO 10303-223:2008 (E)

```

#4719=SHAPE_DEFINING_RELATIONSHIP('', 'vent reference usage', #4730, #4710);
#4720=SHAPE_DEFINING_RELATIONSHIP('', 'choke reference usage', #4750, #4710);
#4721=SHAPE_DEFINING_RELATIONSHIP('', 'filter reference usage', #4300, #4710);
#4728=SHAPE_DEFINING_RELATIONSHIP('', 'sprue reference usage', #4200, #4710);

/* ----- VENT -----*/

#4730=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  GATING_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.)
  VENT());
#4732=PRODUCT_DEFINITION_SHAPE('', '', #4730);

#4734=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4739), #34);
#4736=PROPERTY_DEFINITION('', '', #4730);
#4737=PROPERTY_DEFINITION_REPRESENTATION(#4736, #3716);
#4738=PROPERTY_DEFINITION_REPRESENTATION(#4736, #4734);
#4739=DESCRIPTIVE_REPRESENTATION_ITEM('type of vent', 'air');

#4740=SHAPE_ASPECT('', 'vent occurrence', #4732, .T.);
#4742=FEATURE_COMPONENT_RELATIONSHIP('', 'vent reference usage', #3202, #4740);

#4746=PROPERTY_DEFINITION_REPRESENTATION(#4736, #4747);
#4747=LOCATION_SHAPE_REPRESENTATION('location', (#595), #34);

/* ----- choke -----*/

#4750=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  CHOKE()
  GATING_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.));
#4752=PRODUCT_DEFINITION_SHAPE('', '', #4750);

#4754=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4759), #34);
#4756=PROPERTY_DEFINITION('', '', #4750);
#4757=PROPERTY_DEFINITION_REPRESENTATION(#4756, #4763);
#4758=PROPERTY_DEFINITION_REPRESENTATION(#4756, #4754);
#4759=DESCRIPTIVE_REPRESENTATION_ITEM('type of vent', 'air');

#4760=SHAPE_ASPECT('', 'choke occurrence', #4752, .T.);
#4762=FEATURE_COMPONENT_RELATIONSHIP('', 'choke dimension usage', #3202, #4760);
#4763=LOCATION_SHAPE_REPRESENTATION('location', (#595), #34);
/* ----- ingate_contact -----*/

#4800=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '')GATING_DESIGN_FEATURE()INGATE_CONTACT()SHAPE_ASPEC
T('', '', #43, .T.));
#4802=PRODUCT_DEFINITION_SHAPE('', '', #4800);

```



```

#4804=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4809, #4811, #4812, #4813,
#4814), #34);
#4806=PROPERTY_DEFINITION('', '', #4800);
#4808=SHAPE_DEFINITION_REPRESENTATION(#4806, #4804);
#4809=DESCRIPTIVE_REPRESENTATION_ITEM('cope or drag', 'cope');
#4811=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('runner transition edge round'));
#4812=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('runner transition fillet'));
#4813=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('pattern transition edge round'));
#4814=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('pattern transition fillet'));

#4815=SHAPE_ASPECT('', 'ingate contact occurrence', #4802, .T.);
#4816=SHAPE_DEFINING_RELATIONSHIP('', 'ingate contact shape
usage', #561, #4815);

#4818=SHAPE_DEFINING_RELATIONSHIP('', 'runner reference usage', #4700, #4815);

#4820=SHAPE_DEFINING_RELATIONSHIP('', 'production pattern definition reference
usage', #5000, #4815);

/* ----- ingate -----*/

#4900=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  GATING_DESIGN_FEATURE()
  INGATE()
  SHAPE_ASPECT('', '', #43, .T.));

#4902=PRODUCT_DEFINITION_SHAPE('', '', #4900);

#4904=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #4909), #34);
#4906=PROPERTY_DEFINITION('', '', #4900);
#4908=PROPERTY_DEFINITION_REPRESENTATION(#4906, #4904);
#4909=DESCRIPTIVE_REPRESENTATION_ITEM('cope or drag', 'cope');

#4910=SHAPE_ASPECT('', 'ingate occurrence', #4902, .T.);

```

ISO 10303-223:2008 (E)

```
#4912=FEATURE_COMPONENT_RELATIONSHIP('', 'ingate size usage', #3202, #4910);

#4916=SHAPE_DEFINING_RELATIONSHIP('', 'ingate contact reference
usage', #4800, #4910);

/* ----- production_pattern_definition -----*/

#5000=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  PRODUCTION_PATTERN_DEFINITION()
  SAND_CAST_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.));

#5002=PRODUCT_DEFINITION_SHAPE('', '', #5000);

#5004=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #5009, #5010, #5011), #34);
#5006=PROPERTY_DEFINITION('', '', #5000);
#5008=PROPERTY_DEFINITION_REPRESENTATION(#5006, #5004);
#5009=DESCRIPTIVE_REPRESENTATION_ITEM('condition', 'good');
#5010=MEASURE_REPRESENTATION_ITEM('number of patterns
mounted', COUNT_MEASURE(1.0), #35);
#5011=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('shrink factor'));

#5014=SHAPE_ASPECT('', 'production pattern definition occurrence', #5002, .T.);
#5016=SHAPE_DEFINING_RELATIONSHIP('', 'casting insert reference
usage', #2000, #5014);

#5020=SHAPE_DEFINING_RELATIONSHIP('', 'chaplet pad reference
usage', #3100, #5014);

#5024=SHAPE_DEFINING_RELATIONSHIP('', 'chill reference usage', #2050, #5014);

#5028=SHAPE_DEFINING_RELATIONSHIP('', 'core print reference
usage', #3800, #5014);

#5032=SHAPE_DEFINING_RELATIONSHIP('', 'loose piece reference
usage', #3000, #5014);

#5036=SHAPE_DEFINING_RELATIONSHIP('', 'pattern plate reference
usage', #3500, #5014);

#5040=SHAPE_DEFINING_RELATIONSHIP('', 'riser reference usage', #4500, #5014);

#5044=SHAPE_DEFINING_RELATIONSHIP('', 'riser contact reference
usage', #4400, #5014);

#5048=SHAPE_DEFINING_RELATIONSHIP('', 'production pattern definition shape
usage', #561, #5014);
```

```

#5052=SHAPE_DEFINING_RELATIONSHIP('', 'production core box reference
usage', #5100, #5014);

#5080=PROPERTY_DEFINITION_REPRESENTATION(#5006, #3812);

#5082=PROPERTY_DEFINITION_REPRESENTATION(#5006, #5083);
#5083=PARTING_SURFACE('cope and drag parting', (#600, #3613, #3614, #3616), #34);

#5084=PROPERTY_DEFINITION_REPRESENTATION(#5006, #5085);
#5085=PARTING_SURFACE('offset parting', (#600, #3613, #3614, #3616), #34);

#5086=PROPERTY_DEFINITION_REPRESENTATION(#5006, #5087);
#5087=IN_FACILITY_LOCATION('offset parting', (#5088, #5089, #5090), #34);

#5088=DESCRIPTIVE_REPRESENTATION_ITEM('building or area', 'good');
#5089=DESCRIPTIVE_REPRESENTATION_ITEM('location code', 'good');
#5090=DESCRIPTIVE_REPRESENTATION_ITEM('sublocation', 'good');

#5096=REPRESENTATION_RELATIONSHIP('', '', #5083, #5139);

#5097=REPRESENTATION_RELATIONSHIP('', '', #5085, #79);

/* ----- production_core_box -----*/

#5100=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  PRODUCTION_CORE_BOX()
  SAND_CAST_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.));

#5102=PRODUCT_DEFINITION_SHAPE('', '', #5100);

#5104=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #5109, #5110, #5111, #5112)
, #34);
#5106=PROPERTY_DEFINITION('', '', #5100);
#5108=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5104);
#5109=DESCRIPTIVE_REPRESENTATION_ITEM('core box type', 'wood');
#5110=DESCRIPTIVE_REPRESENTATION_ITEM('core vent type', 'wood');
#5111=MEASURE_REPRESENTATION_ITEM('number of
impressions', COUNT_MEASURE(1.0), #35);
#5112=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('shrink factor'));

#5113=MATERIAL_DESIGNATION('urethane', (#5100));

#5114=SHAPE_ASPECT('', 'production core box occurrence', #5102, .T.);
#5115=SHAPE_DEFINING_RELATIONSHIP('', 'casting insert reference
usage', #2000, #5114);

```

ISO 10303-223:2008 (E)

```

#5116=SHAPE_DEFINING_RELATIONSHIP('', 'chaplet          pad          reference
usage', #3100, #5114);
#5117=SHAPE_DEFINING_RELATIONSHIP('', 'core reference usage', #3700, #5114);
#5118=SHAPE_DEFINING_RELATIONSHIP('', 'loose          piece          reference
usage', #3000, #5114);
#5119=SHAPE_DEFINING_RELATIONSHIP('', 'core          print          reference
usage', #3800, #5114);
#5120=FEATURE_COMPONENT_RELATIONSHIP('', 'production      core      box      size
usage', #3202, #5114);
#5121=FEATURE_COMPONENT_RELATIONSHIP('', 'guide pin size usage', #3202, #5114);
#5122=FEATURE_COMPONENT_RELATIONSHIP('', 'outside size usage', #3202, #5114);
#5123=SHAPE_DEFINING_RELATIONSHIP('', 'investment      area      shape
usage', #561, #5114);
#5124=SHAPE_DEFINING_RELATIONSHIP('', 'impression shape usage', #561, #5114);
#5125=FEATURE_COMPONENT_RELATIONSHIP('', 'vent size usage', #3202, #5114);

#5126=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5127);
#5127=DRAFTED_SURFACE('', (#600, #3813), #34);

#5128=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5129);
#5129=IN_FACILITY_LOCATION('tooling location', (#5088, #5089, #5090), #34);

#5130=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5131);
#5131=PARTING_SURFACE('cope and drag parting', (#600, #3613, #3614, #3616), #34);

#5132=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5133);
#5133=PARTING_SURFACE('offset parting', (#600, #3613, #3614, #3616), #34);

#5134=REPRESENTATION_RELATIONSHIP('', '', #5131, #79);
#5135=REPRESENTATION_RELATIONSHIP('', '', #5133, #5139);
#5136=REPRESENTATION_RELATIONSHIP('', '', #5127, #79);

#5138=PROPERTY_DEFINITION_REPRESENTATION(#5106, #5139);
#5139=PLANAR_SHAPE_REPRESENTATION('vent placement', (#137), #34);

#5145=MACHINE_ALLOWANCE('', '', #5100);
#5146=PROPERTY_DEFINITION_REPRESENTATION(#5145, #8082);
#5147=PROPERTY_DEFINITION_RELATIONSHIP('', '', #5145, #1062);

/* ----- EQUIPMENT -----*/

#5180=CORE_EQUIPMENT('', '', (#5184), #5182);
#5182=ACTION_RESOURCE_TYPE('');
#5184=ACTION_METHOD('', '', '', '');
#5186=APPLIED_ACTION_METHOD_ASSIGNMENT(#5184, #5188, (#5100));
#5188=ACTION_METHOD_ROLE('', '');

#5190=RESOURCE_PROPERTY('', '', #5180);
#5191=RESOURCE_PROPERTY_REPRESENTATION('', '', #5190, #9035);

#5192=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #5180);

```

```

#5193=RESOURCE_PROPERTY('', '#5180');
#5194=RESOURCE_PROPERTY_REPRESENTATION('location', '#5193', '#5087');

/* ----- stop_pin -----*/

#5200=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  SHAPE_ASPECT('', '#43', .T.)
  STOP_PIN());
#5202=PRODUCT_DEFINITION_SHAPE('', '#5200');

#5204=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#5206=PROPERTY_DEFINITION('', '#5200');
#5208=PROPERTY_DEFINITION_REPRESENTATION(#5206, #5204);

#5210=SHAPE_ASPECT('', 'stop pin occurrence', #5202, .T.);
#5212=FEATURE_COMPONENT_RELATIONSHIP('', 'stop pin size usage', #3202, #5210);

/* ----- reset_or_push_pin -----*/

#5300=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  RESET_OR_PUSH_PIN()
  SHAPE_ASPECT('', '#43', .T.));
#5302=PRODUCT_DEFINITION_SHAPE('', '#5300');

#5304=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #5309), #34);
#5306=PROPERTY_DEFINITION('', '#5300');
#5308=PROPERTY_DEFINITION_REPRESENTATION(#5306, #5304);
#5309=DESCRIPTIVE_REPRESENTATION_ITEM('spring description', 'wood');

#5310=SHAPE_ASPECT('', 'reset or push pin occurrence', #5302, .T.);
#5312=FEATURE_COMPONENT_RELATIONSHIP('', 'reset or push pin dimensions size
usage', #3202, #5310);

/* ----- pin_tip -----*/

#5400=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  PIN_TIP()
  SHAPE_ASPECT('', '#43', .T.));
#5402=PRODUCT_DEFINITION_SHAPE('', '#5400');

#5404=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#5406=PROPERTY_DEFINITION('', '#5400');

```

ISO 10303-223:2008 (E)

```
#5408=PROPERTY_DEFINITION_REPRESENTATION(#5406,#5404);

#5446=SHAPE_ASPECT('', 'pin tip occurrence', #5402, .T.);
#5448=SHAPE_DEFINING_RELATIONSHIP('', 'pin tip shape usage', #561, #5446);

/* ----- ejector_pin -----*/

#5500=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  EJECTOR_PIN()
  SHAPE_ASPECT('', '', #43, .T.));

#5502=PRODUCT_DEFINITION_SHAPE('', '', #5500);

#5504=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#5506=PROPERTY_DEFINITION('', '', #5500);
#5508=PROPERTY_DEFINITION_REPRESENTATION(#5506, #5504);

#5510=SHAPE_ASPECT('', 'ejector pin occurrence', #5502, .T.);
#5512=FEATURE_COMPONENT_RELATIONSHIP('', 'ejector pin dimension
usage', #3202, #5510);

#5544=SHAPE_DEFINING_RELATIONSHIP('', 'pin tip reference usage', #5400, #5510);
#5545=SHAPE_DEFINING_RELATIONSHIP('', 'thread reference usage', #31800, #5510);

/* ----- ejector_plate-----*/

#5600=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  EJECTOR_PLATE()
  SHAPE_ASPECT('', '', #43, .T.));

#5602=PRODUCT_DEFINITION_SHAPE('', '', #5600);

#5604=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #5609), #34);
#5606=PROPERTY_DEFINITION('', '', #5600);
#5608=PROPERTY_DEFINITION_REPRESENTATION(#5606, #5604);
#5609=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('plate height'));
#5610=SHAPE_ASPECT('', 'ejector plate occurrence', #5602, .T.);
#5612=FEATURE_COMPONENT_RELATIONSHIP('', 'ejector plate dimension
usage', #3202, #5610);

#5619=MATERIAL_DESIGNATION('', (#5600));
```

```

#5621=SHAPE_DEFINING_RELATIONSHIP('', 'ejector pin reference
usage', #5500, #5610);

#5623=SHAPE_DEFINING_RELATIONSHIP('', 'stop pin reference usage', #5200, #5610);

#5625=SHAPE_DEFINING_RELATIONSHIP('', 'reset or push pin reference
usage', #5300, #5610);

/* ----- ejector_box -----*/

#5700=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_BOX()
  EJECTOR_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.));

#5702=PRODUCT_DEFINITION_SHAPE('', '', #5700);

#5704=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#5706=PROPERTY_DEFINITION('', '', #5700);
#5708=PROPERTY_DEFINITION_REPRESENTATION(#5706, #5704);

#5724=SHAPE_ASPECT('', 'ejector box occurrence', #5702, .T.);
#5725=SHAPE_DEFINING_RELATIONSHIP('', 'ejector plate reference
usage', #5600, #5724);

#5774=FEATURE_COMPONENT_RELATIONSHIP('', 'ejector box size
usage', #3202, #5724);

#5785=SHAPE_DEFINING_RELATIONSHIP('', 'ejector box die reference
usage', #6500, #5724);

/* ----- ejector_system -----*/

#5900=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  EJECTOR_DESIGN_FEATURE()
  EJECTOR_SYSTEM()
  SHAPE_ASPECT('', '', #43, .T.));

#5902=PRODUCT_DEFINITION_SHAPE('', '', #5900);

#5904=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #5909), #34);
#5906=PROPERTY_DEFINITION('', '', #5900);
#5908=PROPERTY_DEFINITION_REPRESENTATION(#5906, #5904);
#5909=DESCRIPTIVE_REPRESENTATION_ITEM('system operated by', 'spring loaded');

#5924=SHAPE_ASPECT('', 'ejector system occurrence', #5902, .T.);

```

ISO 10303-223:2008 (E)

```
#5925=SHAPE_DEFINING_RELATIONSHIP('', 'ejector          system          reference
usage', #5600, #5924);

/* ----- slide -----*/

#6000=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.)
  SLIDE());
#6002=PRODUCT_DEFINITION_SHAPE('', '', #6000);

#6004=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #6014), #34);
#6006=PROPERTY_DEFINITION('', '', #6000);
#6008=PROPERTY_DEFINITION_REPRESENTATION(#6006, #6004);

#6010=SHAPE_ASPECT('', 'slide occurrence', #6002, .T.);
#6012=SHAPE_DEFINING_RELATIONSHIP('', 'slide shape usage', #3202, #6010);
#6013=MATERIAL_DESIGNATION('wood', (#6000));
#6014=DESCRIPTIVE_REPRESENTATION_ITEM('slide activation method', '');
/* ----- die_clamping -----*/

#6100=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_CLAMPING()
  DIE_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.) );
#6102=PRODUCT_DEFINITION_SHAPE('', '', #6100);

#6104=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #6109), #34);
#6106=PROPERTY_DEFINITION('', '', #6100);
#6108=PROPERTY_DEFINITION_REPRESENTATION(#6106, #6104);
#6109=AXIS2_PLACEMENT_3D('clamping position', #1368, #1369, #1370);

/* ----- shot_sleeve -----*/

#6200=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.)
  SHOT_SLEEVE());
#6202=PRODUCT_DEFINITION_SHAPE('', '', #6200);

#6204=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#6206=PROPERTY_DEFINITION('', '', #6200);
#6208=PROPERTY_DEFINITION_REPRESENTATION(#6206, #6204);

#6210=SHAPE_ASPECT('', 'shot sleeve occurrence', #6202, .T.);
#6212=FEATURE_COMPONENT_RELATIONSHIP('', 'shot          sleeve          shape
usage', #3202, #6210);
```



```

/* ----- secondary_tooling -----*/

#6300=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  DIE_DESIGN_FEATURE()
  SECONDARY_TOOLING()
  SHAPE_ASPECT('','',#43,.T.) );
#6302=PRODUCT_DEFINITION_SHAPE('','',#6300);

#6304=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1371,#6309),#34);
#6306=PROPERTY_DEFINITION('','',#6300);
#6308=PROPERTY_DEFINITION_REPRESENTATION(#6306,#6304);
#6309=DESCRIPTIVE_REPRESENTATION_ITEM('type of tooling','wood');

/* ----- cooling_port -----*/

#6400=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  COOLING_PORT()
  DIE_DESIGN_FEATURE()
  SHAPE_ASPECT('','',#43,.T.) );
#6402=PRODUCT_DEFINITION_SHAPE('','',#6400);

#6404=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1371,#6409),#34);
#6406=PROPERTY_DEFINITION('','',#6400);
#6408=PROPERTY_DEFINITION_REPRESENTATION(#6406,#6404);
#6409=DESCRIPTIVE_REPRESENTATION_ITEM('male or female','male');

#6410=SHAPE_ASPECT('','cooling port occurrence',#6402,.T.);
#6412=FEATURE_COMPONENT_RELATIONSHIP('','cooling port size
usage',#3202,#6410);

#6468=PROPERTY_DEFINITION_REPRESENTATION(#6406,#6469);
#6469=PLANAR_SHAPE_REPRESENTATION('port location',( #137),#34);

/* ----- production_die_mould -----*/

#6500=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  DIE_DESIGN_FEATURE()
  PRODUCTION_DIE_MOULD()
  PRODUCTION_TOOL()
  SHAPE_ASPECT('','',#43,.T.) );
#6502=PRODUCT_DEFINITION_SHAPE('','',#6500);

#6504=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1371,#6507,#6508,#6532,#6533)
,#34);
#6505=PROPERTY_DEFINITION('','',#6500);
#6506=PROPERTY_DEFINITION_REPRESENTATION(#6505,#6504);

```

## ISO 10303-223:2008 (E)

```
#6507=DESCRIPTIVE_REPRESENTATION_ITEM('male or female','male mould');
#6508=MEASURE_REPRESENTATION_ITEM('number of
cavities',COUNT_MEASURE(1.0),#35);

#6509=SHAPE_ASPECT('','production tool occurrence',#6502,.T.);
#6510=FEATURE_COMPONENT_RELATIONSHIP('','guide pin shape usage',#3202,#6509);
#6511=SHAPE_DEFINING_RELATIONSHIP('','die core reference usage',#6600,#6509);
#6512=SHAPE_DEFINING_RELATIONSHIP('','ejector system reference
usage',#5900,#6509);
#6513=SHAPE_DEFINING_RELATIONSHIP('','production core box reference
usage',#5100,#6509);

#6515=SHAPE_DEFINING_RELATIONSHIP('','gating system reference
usage',#4000,#6528);
#6516=SHAPE_DEFINING_RELATIONSHIP('','core reference usage',#6600,#6528);

#6517=MACHINE_ALLOWANCE('',',',#6500);

#6518=PROPERTY_DEFINITION_REPRESENTATION(#6505,#6519);
#6519=PARTING_SURFACE('male or female parting',(#600,#3613,#3614,#3616),#34);

#6520=PROPERTY_DEFINITION_REPRESENTATION(#6505,#6521);
#6521=PARTING_SURFACE('offset parting',(#600,#3613,#3614,#3616),#34);

#6522=REPRESENTATION_RELATIONSHIP('',',',#6519,#5139);

#6523=REPRESENTATION_RELATIONSHIP('',',',#6521,#79);

#6524=SHAPE_DEFINING_RELATIONSHIP('','riser reference usage',#4500,#6509);
#6525=SHAPE_DEFINING_RELATIONSHIP('','vent reference usage',#4730,#6509);
#6526=FEATURE_COMPONENT_RELATIONSHIP('','production tool outside shape
usage',#3202,#6509);

#6528=SHAPE_ASPECT('','production die mould occurrence',#6502,.T.);
#6529=SHAPE_DEFINING_RELATIONSHIP('','production mould impression shape
usage',#3202,#6509);
#6530=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4500,#6509);
#6531=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4400,#6509);

#6532=DESCRIPTIVE_REPRESENTATION_ITEM('die type','');
#6533=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
  QUALIFIED_REPRESENTATION_ITEM((#1483,#1484,#1485))
  REPRESENTATION_ITEM('shrink factor'));

#6534=PROPERTY_DEFINITION_REPRESENTATION(#6505,#6535);
#6535=DRAFTED_SURFACE('',(#600,#3813),#34);

#6537=REPRESENTATION_RELATIONSHIP('',',',#6535,#5139);
```

```

#6540=SHAPE_DEFINING_RELATIONSHIP('', 'production tool impression shape
usage', #561, #6509);

#6569=PROPERTY_DEFINITION_REPRESENTATION(#6505, #6570);
#6570=IN_FACILITY_LOCATION('offset parting', (#5088, #5089, #5090), #34);

#6572=FEATURE_COMPONENT_RELATIONSHIP('', 'production die mould outside shape
usage', #3202, #6528);
#6574=FEATURE_COMPONENT_RELATIONSHIP('', 'production die mould dimension
usage', #3232, #6528);
#6576=SHAPE_DEFINING_RELATIONSHIP('', 'production die mould outside shape
usage', #561, #6528);
#6578=SHAPE_DEFINING_RELATIONSHIP('', 'slide reference usage', #6000, #6528);
#6580=SHAPE_DEFINING_RELATIONSHIP('', 'core for die reference
usage', #5100, #6528);

/* ----- die_core -----*/

#6600=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_CORE()
  DIE_DESIGN_FEATURE()
  SHAPE_ASPECT('', '', #43, .T.) );
#6602=PRODUCT_DEFINITION_SHAPE('', '', #6600);

#6604=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #6609, #6533), #34);
#6606=PROPERTY_DEFINITION('', '', #6600);
#6608=PROPERTY_DEFINITION_REPRESENTATION(#6606, #6604);
#6609=DESCRIPTIVE_REPRESENTATION_ITEM('core type', 'sand core');

#6610=PROPERTY_DEFINITION_REPRESENTATION(#6606, #6611);
#6611=DRAFTED_SURFACE('', (#600, #3813), #34);

#6612=SHAPE_ASPECT('', 'die core occurrence', #6602, .T.);
#6613=FEATURE_COMPONENT_RELATIONSHIP('', 'guide pin size usage', #3202, #6612);

#6614=PROPERTY_DEFINITION_REPRESENTATION(#6606, #6615);
#6615=PARTING_SURFACE('', (#600, #3613, #3614, #3616), #34);

#6617=REPRESENTATION_RELATIONSHIP('', '', #6615, #5139);
#6618=REPRESENTATION_RELATIONSHIP('', '', #6611, #79);

#6648=SHAPE_DEFINING_RELATIONSHIP('', 'die core area shape usage', #561, #6612);
#6649=MATERIAL_DESIGNATION('', (#6600));

/* ----- production_die_cast_mould -----*/

#6700=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_DESIGN_FEATURE()

```

ISO 10303-223:2008 (E)

```

        PRODUCTION_DIE_CAST_MOULD()
        PRODUCTION_TOOL()
        SHAPE_ASPECT('', '#43, .T. ');
#6702=PRODUCT_DEFINITION_SHAPE('', '#6700);

#6704=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #6709, #6710, #6711, #6713,
#6749, #6750, #6533), #34);
#6706=PROPERTY_DEFINITION('', '#6700);
#6708=PROPERTY_DEFINITION_REPRESENTATION(#6706, #6704);
#6709=DESCRIPTIVE_REPRESENTATION_ITEM('cooling requirement', 'true');
#6710=DESCRIPTIVE_REPRESENTATION_ITEM('insert', 'unit');
#6711=DESCRIPTIVE_REPRESENTATION_ITEM('port connection type', '');
#6713=DESCRIPTIVE_REPRESENTATION_ITEM('die type', 'cold chamber');
#6719=MATERIAL_DESIGNATION('', (#6700));

#6720=SHAPE_ASPECT('', 'production die cast mould occurrence', #6702, .T.);
#6721=SHAPE_DEFINING_RELATIONSHIP('', 'cooling port reference
usage', #6400, #6720);
#6723=FEATURE_COMPONENT_RELATIONSHIP('', 'production die cast mould dimension
usage', #3202, #6720);
#6725=SHAPE_DEFINING_RELATIONSHIP('', 'gate runner reference
usage', #4900, #6720);
#6727=SHAPE_DEFINING_RELATIONSHIP('', 'secondary tooling reference
usage', #6300, #6720);
#6729=SHAPE_DEFINING_RELATIONSHIP('', 'production die cast mould internal
shape usage', #561, #6720);
#6731=SHAPE_DEFINING_RELATIONSHIP('', 'sprue reference usage', #4200, #6720);
#6733=SHAPE_DEFINING_RELATIONSHIP('', 'slide reference usage', #6000, #6720);
#6735=SHAPE_DEFINING_RELATIONSHIP('', 'movable die clamping reference
usage', #6100, #6720);
#6737=SHAPE_DEFINING_RELATIONSHIP('', 'stationary die clamping reference
usage', #6100, #6720);
#6739=SHAPE_DEFINING_RELATIONSHIP('', 'shot sleeve reference
usage', #6200, #6720);

#6749=DESCRIPTIVE_REPRESENTATION_ITEM('male or female', 'male mould');
#6750=MEASURE_REPRESENTATION_ITEM('number of
cavities', COUNT_MEASURE(1.0), #35);

#6751=SHAPE_ASPECT('', 'production tool occurrence', #6702, .T.);
#6752=FEATURE_COMPONENT_RELATIONSHIP('', 'guide pin shape usage', #3202, #6751);
#6753=SHAPE_DEFINING_RELATIONSHIP('', 'die core reference usage', #6600, #6751);
#6754=SHAPE_DEFINING_RELATIONSHIP('', 'ejector system reference
usage', #5900, #6751);
#6755=SHAPE_DEFINING_RELATIONSHIP('', 'production core box reference
usage', #5100, #6751);
#6556=FEATURE_COMPONENT_RELATIONSHIP('', 'production tool outside shape
usage', #3202, #6751);

#6761=MACHINE_ALLOWANCE('', '#6700);

#6762=PROPERTY_DEFINITION_REPRESENTATION(#6706, #6763);

```

```

#6763=PARTING_SURFACE('male or female parting',(#600,#3613,#3614,#3616),#34);

#6764=PROPERTY_DEFINITION_REPRESENTATION(#6706,#6765);
#6765=PARTING_SURFACE('offset parting',(#600,#3613,#3614,#3616),#34);

#6766=PROPERTY_DEFINITION_REPRESENTATION(#6706,#6767);
#6767=DRAFTED_SURFACE('',(#600,#3813),#34);

#6768=REPRESENTATION_RELATIONSHIP('',',',#6763,#79);
#6769=REPRESENTATION_RELATIONSHIP('',',',#6765,#5139);
#6770=REPRESENTATION_RELATIONSHIP('',',',#6767,#5139);

#6780=SHAPE_DEFINING_RELATIONSHIP('','production tool impression shape
usage',#3202,#6751);
#6781=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4500,#6751);
#6782=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4400,#6751);

/* ----- sand_casting_tooling -----*/

#6800=(CASTING_PRODUCT_DEFINITION()
  MASTER_PATTERN()
  PRODUCT_DEFINITION('',',',#6840,#6841)
  SAND_CASTING_TOOLING());

#6802=PRODUCT_DEFINITION_RELATIONSHIP('',',',',',#42,#6800);

#6838=PRODUCT_CONTEXT('Cast Part Schema',#36,'Casting');
#6839=PRODUCT('',',',',',(#38));
#6840=PRODUCT_DEFINITION_FORMATION('',',',#6839);
#6841=PRODUCT_DEFINITION_CONTEXT('function definition',#36,'cast part');

#6844=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6850),#34);
#6846=PROPERTY_DEFINITION('',',',#6800);
#6848=PROPERTY_DEFINITION_REPRESENTATION(#6846,#6844);
#6850=DESCRIPTIVE_REPRESENTATION_ITEM('shrink factor for master
pattern','good stuff');

#6863=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0),#35);
#6864=PRODUCT_DEFINITION('','STEEL',#40,#41);
#6865=MATERIAL_DESIGNATION('STEEL',(#6864));
#6866=MAKE_FROM_USAGE_OPTION('1','material definition','raw
material',#6800,#6864,1,'raw material',#6863);

#6883=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0),#35);
#6884=PRODUCT_DEFINITION('','urethane',#40,#41);
#6885=MATERIAL_DESIGNATION('urethane',(#6884));
#6886=MAKE_FROM_USAGE_OPTION('1','tooling material','raw
material',#6800,#6884,1,'raw material',#6883);

#6892=PRODUCT_DEFINITION_SHAPE('',',',#6800);

```

ISO 10303-223:2008 (E)

```

#6893=SHAPE_ASPECT('', 'sand casting tooling occurrence', #6892, .T.);
#6894=SHAPE_DEFINING_RELATIONSHIP('', 'production pattern definition reference
usage', #5000, #6893);

#6895=APPLIED_DATE_ASSIGNMENT(#6896, #6897, (#6800, #6900, #7000, #7100, #7700));
#6896=DATE(2006);
#6897=DATE_ROLE('date last used');

#6899=MACHINE_ALLOWANCE('', '', #6800);

/* ----- die_casting_tooling -----*/

#6900=(CASTING_PRODUCT_DEFINITION()
      DIE_CAST_MASTER()
      DIE_CAST_TOOLING()
      PRODUCT_DEFINITION('', '', #6940, #6941));

#6902=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #42, #6900);

#6938=PRODUCT_CONTEXT('Cast Part Schema', #36, 'Casting');
#6939=PRODUCT('', '', '', (#38));
#6940=PRODUCT_DEFINITION_FORMATION('', '', #6939);
#6941=PRODUCT_DEFINITION_CONTEXT('function definition', #36, 'cast part');

#6944=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6949), #34);
#6946=PROPERTY_DEFINITION('', '', #6900);
#6948=PROPERTY_DEFINITION_REPRESENTATION(#6946, #6944);
#6949=(LENGTH_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
      QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
      REPRESENTATION_ITEM('master shrink factor'));
#6950=PROPERTY_DEFINITION_REPRESENTATION(#6946, #6951);
#6951=IN_FACILITY_LOCATION('tooling location', (#5088, #5089, #5090), #34);

#6963=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#6964=PRODUCT_DEFINITION('', 'STEEL', #40, #41);
#6965=MATERIAL_DESIGNATION('STEEL', (#6964));
#6966=MAKE_FROM_USAGE_OPTION('1', 'material definition', 'raw
material', #6900, #6964, 1, 'raw material', #6963);

#6983=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#6984=PRODUCT_DEFINITION('', 'urethane', #40, #41);
#6985=MATERIAL_DESIGNATION('urethane', (#6984));
#6986=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #6900, #6984, 1, 'raw material', #6983);

#6987=PRODUCT_DEFINITION_SHAPE('', '', #6900);

#6993=SHAPE_ASPECT('', 'die cast tooling occurrence', #6987, .T.);
#6994=SHAPE_DEFINING_RELATIONSHIP('', 'production die cast mould reference
usage', #6700, #6993);

```

```

#6995=SHAPE_DEFINING_RELATIONSHIP('', 'production die cast mould reference
usage', #6700, #6993);
#6996=SHAPE_DEFINING_RELATIONSHIP('', 'production die cast mould female
reference usage', #6700, #6993);
#6998=SHAPE_DEFINING_RELATIONSHIP('', 'production die cast mould male
reference usage', #6700, #6993);

#6999=MACHINE_ALLOWANCE('', '', #6900);

/* ----- die_mould_tooling -----*/

#7000=(CASTING_PRODUCT_DEFINITION()
  DIE_MASTER()
  DIE_MOULD_TOOLING()
  PRODUCT_DEFINITION('', '', #7040, #7041));

#7002=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #42, #7000);

#7038=PRODUCT_CONTEXT('Cast Part Schema', #36, 'Casting');
#7039=PRODUCT('', '', '', (#38));
#7040=PRODUCT_DEFINITION_FORMATION('', '', #7039);
#7041=PRODUCT_DEFINITION_CONTEXT('function definition', #36, 'cast part');

#7044=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7049), #34);
#7046=PROPERTY_DEFINITION('', '', #7000);
#7048=PROPERTY_DEFINITION_REPRESENTATION(#7046, #7044);
#7049=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('master shrink factor'));

#7050=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#15030, #7051, (#7000));
#7051=PERSON_AND_ORGANIZATION_ROLE('tooling location');

#7063=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#7064=PRODUCT_DEFINITION('', 'STEEL', #40, #41);
#7065=MATERIAL_DESIGNATION('STEEL', (#7064));
#7066=MAKE_FROM_USAGE_OPTION('1', 'material definition', 'raw
material', #7000, #7064, 1, 'raw material', #7063);

#7083=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#7084=PRODUCT_DEFINITION('', 'urethane', #40, #41);
#7085=MATERIAL_DESIGNATION('urethane', (#7084));
#7086=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #7000, #7084, 1, 'raw material', #7083);

#7087=PRODUCT_DEFINITION_SHAPE('', '', #7000);

#7091=SHAPE_ASPECT('', 'die mould tooling occurrence', #7087, .T.);
#7092=SHAPE_DEFINING_RELATIONSHIP('', 'production die mould reference
usage', #6500, #7091);

```

ISO 10303-223:2008 (E)

```

#7094=SHAPE_DEFINING_RELATIONSHIP('', 'production die mould reference
usage', #6500, #7091);
#7095=SHAPE_ASPECT('', 'die master occurrence', #7087, .T.);
#7096=SHAPE_DEFINING_RELATIONSHIP('', 'production die mould female reference
usage', #6500, #7095);
#7098=SHAPE_DEFINING_RELATIONSHIP('', 'production die mould male reference
usage', #6500, #7095);

#7099=MACHINE_ALLOWANCE('', '', #7000);

/* ----- core_box_tooling and core_master -----*/

#7100=(CASTING_PRODUCT_DEFINITION()
  CORE_BOX_TOOLING()
  CORE_MASTER()
  PRODUCT_DEFINITION('', '', #7140, #7141));

#7102=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #42, #7100);

#7104=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7109), #34);
#7106=PROPERTY_DEFINITION('', '', #7100);
#7108=PROPERTY_DEFINITION_REPRESENTATION(#7106, #7104);
#7109=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('core box master shrink factor'));

#7138=PRODUCT_CONTEXT('Cast Part Schema', #36, 'Casting');
#7139=PRODUCT('', '', '', (#38));
#7140=PRODUCT_DEFINITION_FORMATION('', '', #7139);
#7141=PRODUCT_DEFINITION_CONTEXT('function definition', #36, 'cast part');

#7186=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #7100, #7084, 1, 'raw material', #7083);

#7192=PRODUCT_DEFINITION_SHAPE('', '', #7100);
#7197=SHAPE_ASPECT('', 'core box tooling occurrence', #7192, .T.);
#7198=SHAPE_DEFINING_RELATIONSHIP('', 'production core box reference
usage', #5100, #7197);

/* ----- INVESTMENT_MOULD -----*/

#7200=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  INVESTMENT_DESIGN_FEATURE()
  INVESTMENT_MOULD()
  SHAPE_ASPECT('', '', #43, .T.));

```



```

#7202=PRODUCT_DEFINITION_SHAPE('', '#7200');

#7204=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #7209, #7210), #34);
#7206=PROPERTY_DEFINITION('', '#7200');
#7208=PROPERTY_DEFINITION_REPRESENTATION(#7206, #7204);
#7209=MEASURE_REPRESENTATION_ITEM('number of
patterns', COUNT_MEASURE(1.0), #35);
#7210=DESCRIPTIVE_REPRESENTATION_ITEM('mould material', 'good stuff');

/* ----- lost_foam_casting_die -----*/

#7300=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', '')
  DIE_DESIGN_FEATURE()
  LOST_FOAM_CASTING_DIE()
  PRODUCTION_TOOL()
  SHAPE_ASPECT('', '#43, .T.));
#7302=PRODUCT_DEFINITION_SHAPE('', '#7300);

#7304=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #7309, #7310, #7335, #7336,
#7337, #7338, #6533), #34);
#7306=PROPERTY_DEFINITION('', '#7300);
#7308=PROPERTY_DEFINITION_REPRESENTATION(#7306, #7304);
#7309=DESCRIPTIVE_REPRESENTATION_ITEM('male or female', 'male mould');
#7310=MEASURE_REPRESENTATION_ITEM('number of
cavities', COUNT_MEASURE(1.0), #35);

#7311=SHAPE_ASPECT('', 'production tool occurrence', #7302, .T.);
#7316=SHAPE_DEFINING_RELATIONSHIP('', 'ejector system reference
usage', #5900, #7311);

#7322=PROPERTY_DEFINITION_REPRESENTATION(#7306, #7323);
#7323=PARTING_SURFACE('male or female parting', (#600, #3613, #3614, #3616), #34);

#7327=SHAPE_DEFINING_RELATIONSHIP('', 'production tool impression shape
usage', #3202, #7311);
#7328=FEATURE_COMPONENT_RELATIONSHIP('', 'production tool outside shape
usage', #3202, #7311);

#7334=REPRESENTATION_RELATIONSHIP('', '#7323, #79);

#7335=DESCRIPTIVE_REPRESENTATION_ITEM('raw bead size', 'good stuff');
#7336=DESCRIPTIVE_REPRESENTATION_ITEM('steam line location', 'good stuff');
#7337=DESCRIPTIVE_REPRESENTATION_ITEM('pattern removal instruction', 'good
stuff');
#7338=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('die wall thickness'));

```

## ISO 10303-223:2008 (E)

```
#7342=PROPERTY_DEFINITION_REPRESENTATION(#7306,#7344);
#7344=PLANAR_SHAPE_REPRESENTATION('steam line location',(#137),#34);

#7352=PROPERTY_DEFINITION_REPRESENTATION(#7306,#7354);
#7354=PLANAR_SHAPE_REPRESENTATION('fill gun location',(#137),#34);

#7355=PROPERTY_DEFINITION_REPRESENTATION(#7306,#7356);
#7356=DRAFTED_SURFACE('',(#600,#3813),#34);

#7357=REPRESENTATION_RELATIONSHIP('',',',#7356,#79);

#7361=SHAPE_ASPECT('','lost foam casting die occurrence',#7302,.T.);
#7362=FEATURE_COMPONENT_RELATIONSHIP('','fill gun size usage',#3202,#7361);
#7363=SHAPE_DEFINING_RELATIONSHIP('','cooling port reference
usage',#6400,#7361);
#7364=SHAPE_DEFINING_RELATIONSHIP('','die clamping reference
usage',#6100,#7361);
#7365=SHAPE_DEFINING_RELATIONSHIP('','flask reference usage',#3350,#7361);
#7366=SHAPE_DEFINING_RELATIONSHIP('','gating design feature reference
usage',#4900,#7361);
#7367=SHAPE_DEFINING_RELATIONSHIP('','slide or die clamping reference
usage',#6000,#7361);

#7370=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#15030,#7371,(#7300));
#7371=PERSON_AND_ORGANIZATION_ROLE('tool stored');

#7372=PROPERTY_DEFINITION_REPRESENTATION(#7306,#9575);

/* ----- sprue_and_runner_mould -----*/

#7400=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('',')
  INVESTMENT_DESIGN_FEATURE()
  SHAPE_ASPECT('',',#43,.T.)
  SPRUE_AND_RUNNER_MOULD());

#7401=PRODUCT_DEFINITION_SHAPE('',',#7400);

#7404=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371),#34);
#7406=PROPERTY_DEFINITION('',',#7400);
#7408=PROPERTY_DEFINITION_REPRESENTATION(#7406,#7404);

#7411=SHAPE_ASPECT('','sprue and runner mould occurrence',#7401,.T.);
#7412=SHAPE_DEFINING_RELATIONSHIP('','runner reference usage',#4700,#7411);
#7414=SHAPE_DEFINING_RELATIONSHIP('','sprue reference usage',#4200,#7411);
#7415=SHAPE_DEFINING_RELATIONSHIP('','ingate reference usage',#4900,#7411);
#7416=SHAPE_DEFINING_RELATIONSHIP('','investment mould reference
usage',#7200,#7411);
```

```

/* ----- production_investment_cast_mould -----
*/

#7500=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  DIE_DESIGN_FEATURE()
  PRODUCTION_INVESTMENT_CAST_MOULD()
  PRODUCTION_TOOL()
  SHAPE_ASPECT('','',#43,.T.) );
#7502=PRODUCT_DEFINITION_SHAPE('','',#7500);

#7504=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1371,#7509,#7510,#7532,#7533)
,#34);
#7506=PROPERTY_DEFINITION('','',#7500);
#7508=PROPERTY_DEFINITION_REPRESENTATION(#7506,#7504);
#7509=DESCRIPTIVE_REPRESENTATION_ITEM('male or female','male mould');
#7510=MEASURE_REPRESENTATION_ITEM('number of
cavities',COUNT_MEASURE(1.0),#35);

#7511=SHAPE_ASPECT('','production tool occurrence',#7502,.T.);
#7512=FEATURE_COMPONENT_RELATIONSHIP('','guide pin shape usage',#3202,#7511);
#7513=SHAPE_DEFINING_RELATIONSHIP('','die core reference usage',#6600,#7511);
#7514=SHAPE_DEFINING_RELATIONSHIP('','ejector system reference
usage',#5900,#7511);
#7515=SHAPE_DEFINING_RELATIONSHIP('','production core box reference
usage',#5100,#7511);
#7518=FEATURE_COMPONENT_RELATIONSHIP('','production tool outside shape
usage',#3202,#7511);

#7521=MACHINE_ALLOWANCE('','',#7500);

#7522=PROPERTY_DEFINITION_REPRESENTATION(#7506,#7523);
#7523=PARTING_SURFACE('male or female parting',( #600,#3613,#3614,#3616),#34);

#7524=PROPERTY_DEFINITION_REPRESENTATION(#7506,#7525);
#7525=PARTING_SURFACE('offset parting',( #600,#3613,#3614,#3616),#34);

#7527=SHAPE_DEFINING_RELATIONSHIP('','production tool impression shape
usage',#3202,#7511);
#7529=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4500,#7511);
#7531=SHAPE_DEFINING_RELATIONSHIP('','riser on mould usage',#4400,#7511);

#7532=DESCRIPTIVE_REPRESENTATION_ITEM('die type','');
#7533=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
  QUALIFIED_REPRESENTATION_ITEM((#1483,#1484,#1485))
  REPRESENTATION_ITEM('shrink factor'));

#7534=REPRESENTATION_RELATIONSHIP('','',#7523,#79);

```

ISO 10303-223:2008 (E)

```
#7537=REPRESENTATION_RELATIONSHIP('', '#7525, #5139);

#7539=REPRESENTATION_RELATIONSHIP('', '#7571, #5139);

#7570=PROPERTY_DEFINITION_REPRESENTATION(#7506, #7571);
#7571=DRAFTED_SURFACE('', (#600, #3813), #34);

#7572=PROPERTY_DEFINITION_REPRESENTATION(#7506, #7573);
#7573=IN_FACILITY_LOCATION('offset parting', (#5088, #5089, #5090), #34);

#7577=SHAPE_ASPECT('', 'production investment cast mould
occurrence', #7502, .T.);
#7578=SHAPE_DEFINING_RELATIONSHIP('', 'slide reference usage', #6000, #7577);
#7574=FEATURE_COMPONENT_RELATIONSHIP('', 'production investment cast mould
outside shape usage', #3202, #7577);
#7576=SHAPE_DEFINING_RELATIONSHIP('', 'production investment cast mould
outside shape usage', #561, #7577);

/* ----- investment casting tooling definition -----
---*/

#7600=(CASTING_FEATURE_DEFINITION()
      CASTING_INSTANCED_FEATURE()
      CHARACTERIZED_OBJECT('', '')
      INVESTMENT_CASTING_TOOLING_DEFINITION()
      INVESTMENT_DESIGN_FEATURE()
      SHAPE_ASPECT('', '#43, .T.));

#7602=PRODUCT_DEFINITION_SHAPE('', '#7600);

#7604=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#7606=PROPERTY_DEFINITION('', '#7600);
#7608=PROPERTY_DEFINITION_REPRESENTATION(#7606, #7604);

#7611=SHAPE_ASPECT('', 'investment casting tooling definition
occurrence', #7602, .T.);
#7612=SHAPE_DEFINING_RELATIONSHIP('', 'sprue and runner mould reference
usage', #7400, #7611);
#7614=SHAPE_DEFINING_RELATIONSHIP('', 'investment mould reference
usage', #7200, #7611);
#7616=SHAPE_DEFINING_RELATIONSHIP('', 'production core box reference
usage', #5100, #7611);
#7617=SHAPE_DEFINING_RELATIONSHIP('', 'production core box reference
usage', #5100, #7611);
#7618=SHAPE_DEFINING_RELATIONSHIP('', 'production investment cast mould
reference usage', #7500, #7611);

#7620=SHAPE_DEFINING_RELATIONSHIP('', 'pattern section shape
usage', #561, #7611);

#7634=APPLIED_ACTION_METHOD_ASSIGNMENT(#7650, #7636, (#7600));
```

```

#7636=ACTION_METHOD_ROLE('', '');

/* ----- assembly -----*/

#7650=      (ACTION_METHOD('', '', '', ''))
           ASSEMBLY()
           CASTING_ACTIVITY()
           MANUFACTURING_ACTIVITY()
           NON_CASTING_ACTIVITY()
           PROCESS_PLAN_ACTIVITY());

#7654=ACTION_PROPERTY('', '', #7650);
#7655=ACTION_PROPERTY_REPRESENTATION('', '', #7654, #7656);
#7656=REPRESENTATION('', (#7657, #7658, #7659, #7660, #7661), #34);
#7657=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#7658=DESCRIPTIVE_REPRESENTATION_ITEM('special instruction', 'good stuff');
#7659=DESCRIPTIVE_REPRESENTATION_ITEM('operation description', 'good stuff');

#7660=DESCRIPTIVE_REPRESENTATION_ITEM('pattern          section          assembly
sequence', 'good stuff');
#7661=DESCRIPTIVE_REPRESENTATION_ITEM('foam pattern coating density', 'good
stuff');

/* ----- investment_casting_tooling -----*/

#7700=(CASTING_PRODUCT_DEFINITION()
       INVESTMENT_CASTING_MASTER()
       INVESTMENT_CASTING_TOOLING()
       PRODUCT_DEFINITION('', '', #7740, #7741) );

#7702=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #42, #7700);

#7703=PRODUCT_DEFINITION_RELATIONSHIP('', 'prototype', '', #7700, #33006);

#7738=PRODUCT_CONTEXT('Cast Part Schema', #36, 'Casting');
#7739=PRODUCT('', '', '', (#38));
#7740=PRODUCT_DEFINITION_FORMATION('', '', #7739);
#7741=PRODUCT_DEFINITION_CONTEXT('function definition', #36, 'cast part');

#7744=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7749, #7750), #34);
#7746=PROPERTY_DEFINITION('', '', #7700);
#7748=PROPERTY_DEFINITION_REPRESENTATION(#7746, #7744);
#7749=DESCRIPTIVE_REPRESENTATION_ITEM('tooling identifier', 'true');
#7750=(LENGTH_MEASURE_WITH_UNIT()
       MEASURE_REPRESENTATION_ITEM()
       MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
       QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
       REPRESENTATION_ITEM('shrink factor for master tooling'));

#7763=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#7764=PRODUCT_DEFINITION('', 'STEEL', #40, #41);
#7765=MATERIAL_DESIGNATION('STEEL', (#7764));

```

ISO 10303-223:2008 (E)

```

#7766=MAKE_FROM_USAGE_OPTION('1','material definition','raw
material',#7700,#7764,1,'raw material',#7763);

#7783=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0),#35);
#7784=PRODUCT_DEFINITION('','urethane',#40,#41);
#7785=MATERIAL_DESIGNATION('urethane',(#7784));
#7786=MAKE_FROM_USAGE_OPTION('1','tooling material','raw
material',#7700,#7784,1,'raw material',#7783);

#7792=PRODUCT_DEFINITION_SHAPE('',',',#7700);

#7793=SHAPE_ASPECT('','investment casting tooling occurrence',#7792,.T.);
#7794=SHAPE_DEFINING_RELATIONSHIP('','investment casting tooling definition
reference usage',#7600,#7793);

/* ----- fixture_tab -----*/

#7800=(CASTING_DESIGN_FEATURE()
  CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('',')
  FIXTURE_TAB()
  SHAPE_ASPECT('',',',#43,.T.));

#7802=PRODUCT_DEFINITION_SHAPE('',',',#7800);

#7803=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7805,#7809),#34);

#7805=AXIS2_PLACEMENT_3D('orientation',#1368,#1369,#1370);
#7806=PROPERTY_DEFINITION('',',',#7800);
#7808=PROPERTY_DEFINITION_REPRESENTATION(#7806,#7803);
#7809=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
  QUALIFIED_REPRESENTATION_ITEM((#1483,#1484,#1485))
  REPRESENTATION_ITEM('draft'));

/* ----- planar rib -----*/

#7900=(CASTING_DESIGN_FEATURE()
  CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','planar')
  RIB()
  SHAPE_ASPECT('',',',#43,.T.));

#7902=PRODUCT_DEFINITION_SHAPE('',',',#7900);

#7904=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#1371,#7909,#7910,#7911,#7912,
#7913,#7914),#34);

```

```

#7906=PROPERTY_DEFINITION('', '#7900');
#7908=PROPERTY_DEFINITION_REPRESENTATION(#7906, #7904);
#7909=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('draft'));
#7910=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('part rib radius'));
#7911=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('rib extent fillet'));
#7912=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('transition to top radius'));
#7913=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('wall to top radius'));
#7914=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('wall to floor fillet'));

#7958=PROPERTY_DEFINITION_REPRESENTATION(#7906, #7959);
#7959=FACE_SHAPE_REPRESENTATION('top', (#108), #34);

#7968=PROPERTY_DEFINITION_REPRESENTATION(#7906, #7969);
#7969=PLANAR_SHAPE_REPRESENTATION('walls', (#137), #34);

#7978=PROPERTY_DEFINITION_REPRESENTATION(#7906, #7979);
#7979=PLANAR_SHAPE_REPRESENTATION('top', (#137), #34);

/* ----- general rib -----*/

#8000=(CASTING_DESIGN_FEATURE()
  CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('', 'complex')
  RIB()
  SHAPE_ASPECT('', '#43, .T.));

```

ISO 10303-223:2008 (E)

```
#8002=PRODUCT_DEFINITION_SHAPE('', '#8000');

#8004=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371, #8009, #8010, #8011, #8012), #34);
#8006=PROPERTY_DEFINITION('', '#8000');
#8008=PROPERTY_DEFINITION_REPRESENTATION(#8006, #8004);
#8009=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('draft'));
#8010=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('part rib fillet'));
#8011=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('wall to top radius'));
#8012=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  QUALIFIED_REPRESENTATION_ITEM((#1483, #1484, #1485))
  REPRESENTATION_ITEM('wall to floor fillet'));

#8058=PROPERTY_DEFINITION_REPRESENTATION(#8006, #8059);
#8059=FACE_SHAPE_REPRESENTATION('walls top floor', (#108), #34);

#8068=PROPERTY_DEFINITION_REPRESENTATION(#8006, #8069);
#8069=PLANAR_SHAPE_REPRESENTATION('walls top floor', (#137), #34);

/* ----- machine_allowance -----*/

#8070=PROPERTY_DEFINITION_REPRESENTATION(#6517, #8082);
#8071=PROPERTY_DEFINITION_REPRESENTATION(#6899, #8082);
#8072=PROPERTY_DEFINITION_REPRESENTATION(#6761, #8082);
#8073=PROPERTY_DEFINITION_REPRESENTATION(#6999, #8082);
#8074=PROPERTY_DEFINITION_REPRESENTATION(#7099, #8082);
#8075=PROPERTY_DEFINITION_REPRESENTATION(#7521, #8082);

#8076=PROPERTY_DEFINITION_RELATIONSHIP('', '#6517, #1062);
#8077=PROPERTY_DEFINITION_RELATIONSHIP('', '#6761, #1062);
#8078=PROPERTY_DEFINITION_RELATIONSHIP('', '#6999, #1062);
#8079=PROPERTY_DEFINITION_RELATIONSHIP('', '#7099, #1062);
#8080=PROPERTY_DEFINITION_RELATIONSHIP('', '#7521, #1062);
#8081=PROPERTY_DEFINITION_RELATIONSHIP('', '#6899, #1062);

#8082=REPRESENTATION('', (#8083), #34);
```



```

#8083=(LENGTH_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
      QUALIFIED_REPRESENTATION_ITEM((#1483,#1484,#1485))
      REPRESENTATION_ITEM('allowance value'));

/* ----- melting_equipment -----*/

#9000=(ACTION_RESOURCE('', '#9010', #9020)
      CASTING_EQUIPMENT()
      MACHINE()
      MELTING_EQUIPMENT());
#9010=ACTION_METHOD('', '#', '#');
#9020=ACTION_RESOURCE_TYPE('');

#9022=ACTION_RESOURCE_RELATIONSHIP('', '#11000', #9000);

#9025=RESOURCE_PROPERTY('', '#9000');
#9030=RESOURCE_PROPERTY_REPRESENTATION('', '#9025', #9035);
#9035=REPRESENTATION('equipment type', (#9040), #34);
#9040=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9042=MACHINE_ELEMENT_RELATIONSHIP('', '#11000', #9000);
#9044=RESOURCE_PROPERTY_REPRESENTATION('location', '#9025', #5087);

/* ----- bankout_frame -----*/

#9100=(ACTION_RESOURCE('', '#9110', #9120)
      BANKOUT_FRAME()
      CASTING_EQUIPMENT()
      CORE_EQUIPMENT()
      MACHINE());

#9110=ACTION_METHOD('', '#', '#');
#9120=ACTION_RESOURCE_TYPE('');

#9122=ACTION_RESOURCE_RELATIONSHIP('', '#11000', #9100);

#9125=RESOURCE_PROPERTY('', '#9100');
#9130=RESOURCE_PROPERTY_REPRESENTATION('', '#9125', #9135);
#9135=REPRESENTATION('equipment type', (#9140), #34);
#9140=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9142=MACHINE_ELEMENT_RELATIONSHIP('', '#11000', #9100);
#9144=RESOURCE_PROPERTY_REPRESENTATION('location', '#9125', #5087);

#9151=RESOURCE_PROPERTY_REPRESENTATION('', '#9125', #9152);
#9152=REPRESENTATION('equipment type', (#9153), #34);
#9153=DESCRIPTIVE_REPRESENTATION_ITEM('material type', ' ');

```

## ISO 10303-223:2008 (E)

```
/* ----- core_setter -----*/

#9200=(ACTION_RESOURCE('', '', (#9210), #9220)
    CASTING_EQUIPMENT()
    CORE_EQUIPMENT()
    CORE_SETTER()
    MACHINE());

#9210=ACTION_METHOD('', '', '', '');
#9220=ACTION_RESOURCE_TYPE('');

#9222=ACTION_RESOURCE_RELATIONSHIP('', '', #11000, #9200);

#9225=RESOURCE_PROPERTY('', '', #9200);
#9230=RESOURCE_PROPERTY_REPRESENTATION('', '', #9225, #9235);
#9235=REPRESENTATION('equipment type', (#9240), #34);
#9240=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9242=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #9200);
#9244=RESOURCE_PROPERTY_REPRESENTATION('location', '', #9225, #5087);

#9251=RESOURCE_PROPERTY_REPRESENTATION('', '', #9225, #9252);
#9252=REPRESENTATION('equipment type', (#9253), #34);
#9253=DESCRIPTIVE_REPRESENTATION_ITEM('moulding machine', ' ');

/* ----- core_bench -----*/

#9300=(ACTION_RESOURCE('', '', (#9310), #9320)
    CASTING_EQUIPMENT()
    CORE_BENCH()
    CORE_EQUIPMENT()MACHINE());

#9310=ACTION_METHOD('', '', '', '');
#9320=ACTION_RESOURCE_TYPE('');

#9322=ACTION_RESOURCE_RELATIONSHIP('', '', #11000, #9300);

#9325=RESOURCE_PROPERTY('', '', #9300);
#9330=RESOURCE_PROPERTY_REPRESENTATION('', '', #9325, #9335);
#9335=REPRESENTATION('equipment type', (#9340), #34);
#9340=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9342=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #9300);
#9344=RESOURCE_PROPERTY_REPRESENTATION('location', '', #9325, #5087);

#9351=RESOURCE_PROPERTY_REPRESENTATION('', '', #9325, #9352);
#9352=REPRESENTATION('equipment type', (#9353, #9354), #34);
#9353=DESCRIPTIVE_REPRESENTATION_ITEM('wax vent', ' ');
```

```

#9354=MEASURE_REPRESENTATION_ITEM('core required',COUNT_MEASURE(1.0),#35);

/* ----- core_conveyer_system ----- */

#9400=(ACTION_RESOURCE('', '#9410', #9420)
    CASTING_EQUIPMENT()
    CORE_CONVEYER_SYSTEM()
    CORE_EQUIPMENT()MACHINE());

#9410=ACTION_METHOD('', '#9410', '#9410');
#9420=ACTION_RESOURCE_TYPE('');

#9422=ACTION_RESOURCE_RELATIONSHIP('', '#9410', #11000, #9400);

#9425=RESOURCE_PROPERTY('', '#9400');
#9430=RESOURCE_PROPERTY_REPRESENTATION('', '#9425', #9435);
#9435=REPRESENTATION('equipment type', (#9440), #34);
#9440=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9442=MACHINE_ELEMENT_RELATIONSHIP('', '#9400', #11000, #9400);
#9444=RESOURCE_PROPERTY_REPRESENTATION('location', '#9425', #5087);

#9451=RESOURCE_PROPERTY_REPRESENTATION('', '#9425', #9452);
#9452=REPRESENTATION('', (#9453, #9454, #9455, #9456, #9457, #9458, #9459, #9460, #9461, #9462, #9463, #9464), #34);
#9453=DESCRIPTIVE_REPRESENTATION_ITEM('cycle time', ' ');
#9454=MEASURE_REPRESENTATION_ITEM('core required',COUNT_MEASURE(1.0),#35);
#9455=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('core box skid height'));
#9456=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('core box skid length'));
#9457=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('core box skid width'));
#9458=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('core box skid diameter'));
#9459=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('core box skid draft'));
#9460=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)

```

ISO 10303-223:2008 (E)

```

        REPRESENTATION_ITEM('core plug height'));
#9461=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
        REPRESENTATION_ITEM('core plug length'));
#9462=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
        REPRESENTATION_ITEM('core plug width'));
#9463=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
        REPRESENTATION_ITEM('core plug diameter'));
#9464=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
        REPRESENTATION_ITEM('core plug draft'));

/* ----- core_machine -----*/

#9500=CORE_MACHINE('', '#9510', #9520);

#9510=ACTION_METHOD('', '#9510', '#9520');
#9520=ACTION_RESOURCE_TYPE('#9510');

#9522=ACTION_RESOURCE_RELATIONSHIP('#9510', '#9520', #11000, #9500);

#9525=RESOURCE_PROPERTY('#9510', '#9520', #9500);
#9530=RESOURCE_PROPERTY_REPRESENTATION('#9510', '#9520', #9525, #9535);
#9535=REPRESENTATION('equipment type', (#9540), #34);
#9540=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', '#9540');

#9542=MACHINE_ELEMENT_RELATIONSHIP('#9510', '#9520', #11000, #9500);
#9544=RESOURCE_PROPERTY_REPRESENTATION('location', '#9510', #9525, #5087);

#9695=MACHINE_ELEMENT_RELATIONSHIP('#9510', '#9520', #9700, #9500);

/* ----- machine_mounting -----*/

#9570=RESOURCE_PROPERTY_REPRESENTATION('#9510', '#9520', #9525, #9575);
#9575=MACHINE_MOUNTING('#9510', (#9588, #9589, #9590, #9591, #9592, #9593, #9594, #9587), #34);

#9576=REPRESENTATION_RELATIONSHIP('#9510', '#9520', #9575, #9577);
#9577=LOCATION_SHAPE_REPRESENTATION('location', (#9595), #34);

#9587=DESCRIPTIVE_REPRESENTATION_ITEM('mount with', 'holes');
#9588=DESCRIPTIVE_REPRESENTATION_ITEM('machine mounting type', 'good');
#9589=DESCRIPTIVE_REPRESENTATION_ITEM('material type', 'good');
#9590=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)

```

```

    REPRESENTATION_ITEM('diameter'));
#9591=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('draft'));
#9592=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('length'));
#9593=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('width'));
#9594=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('height'));

/* ----- core_box_vent -----*/

#9670=RESOURCE_PROPERTY_REPRESENTATION('','','#9525,#9675);
#9675=CORE_BOX_VENT('',( #9688,#9689,#9690,#9691,#9692,#9693,#9694),#34);

#9686=REPRESENTATION_RELATIONSHIP('','','#9675,#9687);
#9687=SHAPE_REPRESENTATION('',( #595),#34);

#9688=DESCRIPTIVE_REPRESENTATION_ITEM('vent type','good');
#9689=DESCRIPTIVE_REPRESENTATION_ITEM('material type','nylon');
#9690=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('diameter'));
#9691=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('draft'));
#9692=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('length'));
#9693=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('width'));
#9694=(LENGTH_MEASURE_WITH_UNIT()
    MEASURE_REPRESENTATION_ITEM()
    MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
    REPRESENTATION_ITEM('height'));

/* ----- ejector_equipment -----*/

#9700=EJECTOR_EQUIPMENT('','',( #9710),#9720);

```

## ISO 10303-223:2008 (E)

```
#9710=ACTION_METHOD('', '', '', '');
#9720=ACTION_RESOURCE_TYPE('');

#9721=APPLIED_ACTION_METHOD_ASSIGNMENT(#9710, #9722, (#5900));
#9722=ACTION_METHOD_ROLE('finishing operation', '');

#9723=ACTION_RESOURCE_RELATIONSHIP('', '', #11000, #9700);

#9725=RESOURCE_PROPERTY('', '', #9700);
#9730=RESOURCE_PROPERTY_REPRESENTATION('', '', #9725, #9735);
#9735=REPRESENTATION('equipment type', (#9740), #34);
#9740=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9742=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #9700);
#9744=RESOURCE_PROPERTY_REPRESENTATION('location', '', #9725, #5087);

/* ----- shakeout -----*/

#9800=SHAKEOUT('', '', (#9810), #9820);

#9810=ACTION_METHOD('', '', '', '');
#9820=ACTION_RESOURCE_TYPE('');

#9822=ACTION_RESOURCE_RELATIONSHIP('', '', #11000, #9800);

#9825=RESOURCE_PROPERTY('', '', #9800);
#9830=RESOURCE_PROPERTY_REPRESENTATION('', '', #9825, #9835);
#9835=REPRESENTATION('equipment type', (#9840), #34);
#9840=DESCRIPTIVE_REPRESENTATION_ITEM('equipment type', ' ');

#9842=MACHINE_ELEMENT_RELATIONSHIP('', '', #11000, #9800);
#9844=RESOURCE_PROPERTY_REPRESENTATION('location', '', #9825, #5087);

/* ----- controller -----*/

#11000=(ACTION_RESOURCE('', '', (#9010), #9020)
    CONTROLLER());

#11020=APPLIED_DOCUMENT_REFERENCE(#11021, '', (#11000));
#11021=PROCESS_PLAN_SPECIFICATION('', '', '', #11022);
#11022=DOCUMENT_TYPE('');
#11023=DOCUMENT_REPRESENTATION_TYPE('digital', #11021);
#11024=DOCUMENT_USAGE_CONSTRAINT(#11021, 'subclass', 'value for subclass');
#11025=DOCUMENT_USAGE_CONSTRAINT(#11021, 'revision', 'value for revision');

/* ----- resource_with_material -----*/

#12010=APPLIED_ACTION_METHOD_ASSIGNMENT(#12089, #12012, (#42));
#12012=ACTION_METHOD_ROLE('', '');
```

```

#12089=ACTION_METHOD('', '', '', '');
#12090=RESOURCE_WITH_MATERIAL('', '', (#12089), #12091);
#12091=ACTION_RESOURCE_TYPE('');
#12092=RESOURCE_PROPERTY('', '', #12090);
#12093=RESOURCE_PROPERTY_REPRESENTATION('', '', #12092, #12094);
#12094=REPRESENTATION('', (#12095), #34);

#12095=MEASURE_REPRESENTATION_ITEM('resource
quantity', COUNT_MEASURE(1.0), #12096);
#12096=CONTEXT_DEPENDENT_UNIT(#5, 'each');

#12097=RESOURCE_REQUIREMENT_TYPE('', '');
#12098=REQUIREMENT_FOR_ACTION_RESOURCE('', '', #12097, (#12099), (#12090));
#12099=PROPERTY_PROCESS('test', '', #13000, '');
#13000=ACTION_METHOD('', '', '', '');
#13001=PROCESS_PROPERTY_ASSOCIATION('', 'resource material', #12099, #13048);
#13044=PRODUCT_DEFINITION('material', '', #40, #41);
#13045=MAKE_FROM_USAGE_OPTION('', '', '', #13044, #42, 1, '', #13046);
#13046=MEASURE_WITH_UNIT(COUNT_MEASURE(1.000000E+000), #13047);
#13047=CONTEXT_DEPENDENT_UNIT(#5, 'COUNT');
#13048=MATERIAL_PROPERTY('', '', #13044);

/* ----- resource_with_representation -----*/

#14010=RESOURCE_WITH_REPRESENTATION('', '', (#15710), #14011);
#14011=ACTION_RESOURCE_TYPE('');
#14012=RESOURCE_PROPERTY_REPRESENTATION('', '', #14013, #14014);
#14013=RESOURCE_PROPERTY('', '', #14010);
#14014=EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS('', (#14048), #34);
#14045=CARTESIAN_POINT('value for location', (0.000000E+000, 0.000000E+000,
0.000000E+000));
#14046=DIRECTION('', (0.000000E+000, 1.000000E+000, 0.000000E+000));
#14047=DIRECTION('', (1.000000E+000, 0.000000E+000, 0.000000E+000));
#14048=AXIS2_PLACEMENT_3D('value for placement', #14045, #14046, #14047);

#14050=RESOURCE_PROPERTY_REPRESENTATION('', '', #14013, #14051);
#14051=REPRESENTATION('', (#14052), #34);
#14052=MEASURE_REPRESENTATION_ITEM('resource
quantity', COUNT_MEASURE(1.0), #14053);
#14053=CONTEXT_DEPENDENT_UNIT(#5, 'each');

#14054=CLASSIFICATION_ROLE('definitional class membership', '');

#14055=APPLIED_LIBRARY_ASSIGNMENT(#14056, #14054, (#14014));
#14056=EXTERNALLY_DEFINED_CLASS(IDENTIFIER(''), #14057, 'library identifier',
'');
#14057=KNOWN_SOURCE(IDENTIFIER(''), 'ISO 13584 library');
#14058=ORGANIZATION_ROLE('library supplier');
#14059=ORGANIZATION('value for BSU.code', '', '');
#14060=APPLIED_ORGANIZATION_ASSIGNMENT(#14059, #14058, (#14057));
#14061=LIBRARY_CLASS_VERSION_ASSIGNMENT('value for Class_BSU version',
#14062, #14057, (#14056));

```

## ISO 10303-223:2008 (E)

```
#14062=IDENTIFICATION_ROLE('class version', '');
#14063=EXTERNALLY_DEFINED_GENERAL_PROPERTY('', '', '', IDENTIFIER(''),
#14057);
#14064=EXTERNALLY_DEFINED_ITEM_RELATIONSHIP('name scope', '', #14063,
#14056);
#14065=GENERAL_PROPERTY_ASSOCIATION('definitional', '', #14063, #14066);
#14066=PROPERTY_DEFINITION('', '', #42);
#14067=PROPERTY_DEFINITION_REPRESENTATION(#14066, #14068);
#14068=REPRESENTATION('property value', (#14070), #34);
#14070=VALUE_REPRESENTATION_ITEM('value for Property_value.value_amount',
LENGTH_MEASURE(2.000000E+000));
#14071=LIBRARY_CLASS_VERSION_ASSIGNMENT('value for Property_BSU version',
#14072, #14057, (#14063));
#14072=IDENTIFICATION_ROLE('property version', '');

#15005=SECURITY_CLASSIFICATION('classification', 'classify product
version', #15006);
#15006=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#15007=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#15005, (#40, #6840, #6940, #70
40, #7140, #7740, #18040, #18140, #30140, #30240));

#15010=APPLIED_APPROVAL_ASSIGNMENT(#15020, (#1531));
#15012=ACTION_RELATIONSHIP('', '', #1531, #15040);
#15013=ACTION_RELATIONSHIP('', '', #1531, #15050);
#15014=ACTION_RELATIONSHIP('', '', #1531, #15060);
#15015=ACTION_RELATIONSHIP('', '', #1531, #15070);
#15016=APPLIED_ACTION_ASSIGNMENT(#1531, (#40, #6840, #6940, #7040, #7140, #7740, #30
140, #30240));

/* ----- approval and date -----*/

#15020=APPROVAL(#15021, '');
#15021=APPROVAL_STATUS('approved');
#15022=APPROVAL_DATE_TIME(#15023, #15020);
#15023=DATE(2006);

/* ----- address, person, organization, and person_in_organization -----*/

#15024=ORGANIZATIONAL_ADDRESS('', '1398', 'Gumbert', '', 'Amelia', 'Ohio',
'45102', 'USA', '', '513-383-8311', 'slovensky@scra.org', '', (#15026), '');
#15026=ORGANIZATION('NGIT', 'Northrop Grumman', '');
#15028=PERSON('1', 'Slovensky', 'Len', ('W'), ('Dr'), (''));
#15030=PERSON_AND_ORGANIZATION(#15028, #15026);

/* ----- digital_technical_data_packagework_order -----*/

#15040=DIRECTED_ACTION('digital technical data package work
order', '', #15042, #15044);
#15042=ACTION_METHOD('', '', '', '');
#15044=ACTION_DIRECTIVE('order id 1234', '', '', '', (#15046));
```



```

#15046=VERSIONED_ACTION_REQUEST('', '', '', '');

/* ----- pedigree creation order -----*/

#15050=DIRECTED_ACTION('pedigree creation order', '', #15052, #15054);
#15052=ACTION_METHOD('', '', '', '');
#15054=ACTION_DIRECTIVE('order id 1234', '', '', '', (#15056));
#15056=VERSIONED_ACTION_REQUEST('', '', '', '');

/* ----- resource_acquisition_order -----*/

#15060=DIRECTED_ACTION('resource acquisition order', '', #15062, #15064);
#15062=ACTION_METHOD('', '', '', '');
#15064=ACTION_DIRECTIVE('order id 1234', '', '', '', (#15066));
#15066=VERSIONED_ACTION_REQUEST('', '', '', '');

/* ----- shop_work_order -----*/

#15070=DIRECTED_ACTION('shop work order', '', #15072, #15074);
#15072=ACTION_METHOD('', '', '', '');
#15074=ACTION_DIRECTIVE('order id 1234', '', '', '', (#15076));
#15076=VERSIONED_ACTION_REQUEST('', '', '', '');

/* ----- requisition -----*/

#15080=DOCUMENT('requisition number 123', 'quantity ordered 1234', 'requisition
deswcription 1234', #15082);
#15082=DOCUMENT_TYPE('die mould stock requisition');
#15084=APPLIED_DATE_ASSIGNMENT(#15086, #15088, (#15080, #15090, #15094, #15098));
#15086=DATE(2006);
#15088=DATE_ROLE('');

#15090=DOCUMENT('requisition number 123', 'quantity ordered 1234', 'requisition
deswcription 1234', #15092);
#15092=DOCUMENT_TYPE('indirect stock requisition');

#15094=DOCUMENT('requisition number 123', 'quantity ordered 1234', 'requisition
deswcription 1234', #15096);
#15096=DOCUMENT_TYPE('material stock requisition');

#15098=DOCUMENT('requisition number 123', 'quantity ordered 1234', 'requisition
deswcription 1234', #15099);
#15099=DOCUMENT_TYPE('pattern tooling requisition');

/* ----- ordered_part -----*/

#15100=(ACTION_ASSIGNMENT(#15180)
    CHARACTERIZED_OBJECT('quantity', 'quantity ordered')
    ORDERED_PART((#40, #19270, #19470)) );

#15167=MEASURE_REPRESENTATION_ITEM('quantity', COUNT_MEASURE(1), #15168);
#15168=CONTEXT_DEPENDENT_UNIT(#5, 'each');

```

ISO 10303-223:2008 (E)

```

#15169=REPRESENTATION('quantity', (#15167), #34);
#15170=PROPERTY_DEFINITION('ordered part property', 'quantity
definition', #15100);
#15171=PROPERTY_DEFINITION_REPRESENTATION(#15170, #15169);

/* ----- customer_order -----*/

#15180=DIRECTED_ACTION('customer order', 'ship with finished
product', #15182, #15183);
#15182=ACTION_METHOD('', '', '', '');
#15183=ACTION_DIRECTIVE('123-CO-234', '', '', 'ship via ground
transportation', (#16060, #16020));

#15185=APPLIED_DATE_ASSIGNMENT(#15186, #15188, (#15180));
#15186=DATE(2006);
#15188=DATE_ROLE('delivery date');
#15189=APPLIED_DOCUMENT_REFERENCE(#18000, '', (#15180));

#15190=ACTION_STATUS('waiting on TDP', #15180);
#15191=ACTION_PROPERTY('', '', #15180);
#15192=ACTION_PROPERTY_REPRESENTATION('', '', #15191, #15193);
#15193=REPRESENTATION('', (#15194, #15195), #34);
#15194=DESCRIPTIVE_REPRESENTATION_ITEM('production status', 'experimental');
#15195=DESCRIPTIVE_REPRESENTATION_ITEM('casting method', 'core presentation');

#15196=ACTION_RELATIONSHIP('', '', #16000, #15180);
#15197=APPLIED_PERSON_AND_ORGANIZATION_ASSIGNMENT(#15030, #15198, (#15180));
#15198=PERSON_AND_ORGANIZATION_ROLE('customer');
#15199=ACTION_RELATIONSHIP('', '', #1531, #15180);

/* ----- inspection_or_test_requirement -----*/

#15200=INSPECTION_OR_TEST_REQUIREMENT('', '', #42);
#15201=REPRESENTATION('quantity', (#15203, #15204, #15205, #15206), #34);
#15202=PROPERTY_DEFINITION_REPRESENTATION(#15200, #15201);
#15203=DESCRIPTIVE_REPRESENTATION_ITEM('validation id', 'good stuff');
#15204=DESCRIPTIVE_REPRESENTATION_ITEM('test name', 'good stuff');
#15205=MEASURE_REPRESENTATION_ITEM('frequency', COUNT_MEASURE(1.0), #35);
#15206=MEASURE_REPRESENTATION_ITEM('sampling run
size', COUNT_MEASURE(1.0), #35);
#15207=PROPERTY_DEFINITION_RELATIONSHIP('', '', #15200, #15300);
#15208=APPLIED_DOCUMENT_REFERENCE(#15210, '', (#15200));

#15210=DOCUMENT('', '', 'production quality requirement standard', #15212);
#15212=DOCUMENT_TYPE('');

/* ----- special_inspection_requirement -----*/

#15300=SPECIAL_INSPECTION_REQUIREMENT('', '', #42);
#15301=REPRESENTATION('quantity', (#15303, #15304, #15305), #34);
#15302=PROPERTY_DEFINITION_REPRESENTATION(#15300, #15301);

```

```

#15303=DESCRIPTIVE_REPRESENTATION_ITEM('inspection      description','presure
test');
#15304=DESCRIPTIVE_REPRESENTATION_ITEM('other testing method','good stuff');
#15305=DESCRIPTIVE_REPRESENTATION_ITEM('performed by customer','TRUE');
#15308=APPLIED_DOCUMENT_REFERENCE(#15310,'',(#15300));

#15310=DOCUMENT('','',',',#15312);
#15312=DOCUMENT_TYPE('inspection requirements');

/* ----- casting_service_data -----*/

#15400=CASTING_SERVICE_DATA('','',#42);
#15401=REPRESENTATION('quantity',(#15403,#15404,#15405,#15406,#15407),#34);
#15402=PROPERTY_DEFINITION_REPRESENTATION(#15400,#15401);
#15403=DESCRIPTIVE_REPRESENTATION_ITEM('comments','good stuff');
#15404=DESCRIPTIVE_REPRESENTATION_ITEM('wear      or      abrasion
lubrication','none');
#15405=DESCRIPTIVE_REPRESENTATION_ITEM('performed by customer','TRUE');
#15406=DESCRIPTIVE_REPRESENTATION_ITEM('subject to wear against','good
stuff');
#15407=DESCRIPTIVE_REPRESENTATION_ITEM('subject to abrasize wear
against','good stuff');

#15408=PROPERTY_DEFINITION_RELATIONSHIP('','',#15400,#15500);
#15409=PROPERTY_DEFINITION_RELATIONSHIP('','',#15400,#15600);
#15410=PROPERTY_DEFINITION_RELATIONSHIP('','',#15400,#15700);
#15411=PROPERTY_DEFINITION_RELATIONSHIP('','',#15400,#15800);

/* ----- corrosion_service_requirement -----*/

#15500=CORROSION_SERVICE('','',#42);
#15501=REPRESENTATION('quantity',(#15503,#15504,#15505,#15506),#34);
#15502=PROPERTY_DEFINITION_REPRESENTATION(#15500,#15501);
#15503=DESCRIPTIVE_REPRESENTATION_ITEM('corrosive environment','good stuff');
#15504=DESCRIPTIVE_REPRESENTATION_ITEM('corrosive material','good stuff');
#15505=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
REPRESENTATION_ITEM('ph value'));
#15506=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.062000000000000),#15)
REPRESENTATION_ITEM('temperature'));

/* ----- elevated_temperature_service -----*/

#15600= ELEVATED_TEMPERATURE_SERVICE('','',#42);
#15601=REPRESENTATION('quantity',(#15603,#15604,#15605,#15606),#34);
#15602=PROPERTY_DEFINITION_REPRESENTATION(#15600,#15601);
#15603=DESCRIPTIVE_REPRESENTATION_ITEM('service description','good stuff');
#15604=(LENGTH_MEASURE_WITH_UNIT()

```

ISO 10303-223:2008 (E)

```

        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
        REPRESENTATION_ITEM('maximum temperature'));
#15605=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
        REPRESENTATION_ITEM('maximum cyclic temperature'));
#15606=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
        REPRESENTATION_ITEM('minimum cyclic temperature'));

/* ----- finishing_and_machining_operation -----*/

#15700= FINISHING_AND_MACHINING_OPERATION('', '#42);
#15701=REPRESENTATION('quantity', (#15703), #34);
#15702=PROPERTY_DEFINITION_REPRESENTATION(#15700, #15701);
#15703=DESCRIPTIVE_REPRESENTATION_ITEM('comments', 'good stuff');

/* ----- machining_activity -----*/

#15710=      (ACTION_METHOD('', '', '', ''))
            ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#15722, #18261))
            CASTING_ACTIVITY()
            MACHINING_ACTIVITY()
            MANUFACTURING_ACTIVITY()
            NON_CASTING_ACTIVITY()
            PROCESS_PLAN_ACTIVITY());

#15711=APPLIED_ACTION_METHOD_ASSIGNMENT(#15710, #15712, (#42, #15700, #30480));
#15712=ACTION_METHOD_ROLE('finishing operation', '');

#15714=ACTION_PROPERTY('', '#15710);
#15715=ACTION_PROPERTY_REPRESENTATION('', '#15714, #15716);
#15716=REPRESENTATION('', (#15717, #15718, #15719), #34);
#15717=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#15718=DESCRIPTIVE_REPRESENTATION_ITEM('special instruction', 'good stuff');
#15719=DESCRIPTIVE_REPRESENTATION_ITEM('operation description', 'good stuff');

#15720=ACTION_RESOURCE('frequency', '', (#15710), #15721);
#15721=ACTION_RESOURCE_TYPE('');
#15722=DOCUMENT('', '#15723);
#15723=DOCUMENT_TYPE('supplemental document');

/* ----- surface_finish -----*/

#15724=      (ACTION_METHOD('', '', '', ''))
            CASTING_ACTIVITY()
            MANUFACTURING_ACTIVITY()
            NON_CASTING_ACTIVITY()
            PROCESS_PLAN_ACTIVITY()

```

```

SURFACE_FINISH();
#15725=ACTION_PROPERTY('', '#15724');
#15726=ACTION_PROPERTY_REPRESENTATION('', '#15725, #15727');
#15727=REPRESENTATION('', (#15728, #15729), #34);
#15728=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#15729=DESCRIPTIVE_REPRESENTATION_ITEM('operation description', 'good stuff');

#15730=MACHINING_ACTIVITY('', '#15730');
#15731=APPLIED_ACTION_METHOD_ASSIGNMENT(#15730, #15732, (#15700, #18142));
#15732=ACTION_METHOD_ROLE('machining operation', '#15732');

#15734=ACTION_PROPERTY('', '#15734');
#15735=ACTION_PROPERTY_REPRESENTATION('', '#15735, #15736');
#15736=REPRESENTATION('', (#15737, #15738), #34);
#15737=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#15738=DESCRIPTIVE_REPRESENTATION_ITEM('operation description', 'good stuff');

/* ----- heat_treat_requirement -----*/

#15740= HEAT_TREAT_REQUIREMENT('', '#15740');
#15741=REPRESENTATION('quantity', (#15743, #15744, #15745, #15746), #34);
#15742=PROPERTY_DEFINITION_REPRESENTATION(#15740, #15741);
#15743=DESCRIPTIVE_REPRESENTATION_ITEM('additional requirement', 'good stuff');
#15744=DESCRIPTIVE_REPRESENTATION_ITEM('softening anneal required', 'TRUE');
#15745=DESCRIPTIVE_REPRESENTATION_ITEM('stress relief required', 'TRUE');
#15746=DESCRIPTIVE_REPRESENTATION_ITEM('process name', 'good stuff');

#15756=ORGANIZATION($, 'RPTS', 'equipment company name');
#15757=ORGANIZATION_ROLE('activity organization id');
#15758=APPLIED_ORGANIZATION_ASSIGNMENT(#15756, #15757, (#15740, #15710));

#15760=PROPERTY_DEFINITION_RELATIONSHIP('', '#15740, #15762');
#15762=PROPERTY_DEFINITION('process property', '#15762');
#15763=PROPERTY_DEFINITION_RELATIONSHIP('', '#15740, #15770');

#15764=APPLIED_DOCUMENT_REFERENCE(#15766, '#15740');

#15766=DOCUMENT('', '#15766');
#15768=DOCUMENT_TYPE('heat_treat_specification');

/* ----- mechanical_property_requirements -----*/

#15770= MECHANICAL_PROPERTY_REQUIREMENTS('', '#15770');
#15771=REPRESENTATION('quantity', (#15773, #15774, #15775, #15776), #34);
#15772=PROPERTY_DEFINITION_REPRESENTATION(#15770, #15771);
#15773=DESCRIPTIVE_REPRESENTATION_ITEM('chemical certification', 'TRUE') analysis
#15774=DESCRIPTIVE_REPRESENTATION_ITEM('mechanical certification', 'TRUE') properties
#15775=DESCRIPTIVE_REPRESENTATION_ITEM('tensile strength', 'good stuff');
#15776=DESCRIPTIVE_REPRESENTATION_ITEM('yeild strength', 'good stuff');

```

ISO 10303-223:2008 (E)

```

#15784=APPLIED_DOCUMENT_REFERENCE(#15786,'',(#15770));

#15786=DOCUMENT('','',',',#15788);
#15788=DOCUMENT_TYPE('requirement specification');
#15789=PROPERTY_DEFINITION_RELATIONSHIP('','',#15770,#13048);

/* ----- mechanical_stress -----*/

#15800= MECHANICAL_STRESS('','',#42);
#15801=REPRESENTATION('quantity',(#15803,#15804,#15806),#34);
#15802=PROPERTY_DEFINITION_REPRESENTATION(#15800,#15801);
#15803=DESCRIPTIVE_REPRESENTATION_ITEM('type of stress','good stuff');
#15804=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
  REPRESENTATION_ITEM('maximum psi design stress'));
#15806=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
  REPRESENTATION_ITEM('percent safety factor required'));

/* ----- reporting_requirement -----*/

#15810=REPORTING_REQUIREMENT('','',#42);
#15816=APPLIED_DATE_ASSIGNMENT(#15817,#15818,(#15810));
#15817=DATE(2006);
#15818=DATE_ROLE('required delivery date');
#15819=PROPERTY_DEFINITION_RELATIONSHIP('','',#15810,#15900);

/* ----- flaskless -----*/

#15830=(CASTING_FEATURE_DEFINITION()
  CASTING_INSTANCED_FEATURE()
  CHARACTERIZED_OBJECT('','')
  FLASKLESS()
  SAND_CAST_DESIGN_FEATURE()
  SHAPE_ASPECT('','',#43,.T.));
#15831=PRODUCT_DEFINITION_SHAPE('','',#15830);

#15835=DESCRIPTIVE_REPRESENTATION_ITEM('description','good stuff');

#15836=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #1371,#15835),#34);
#15838=PROPERTY_DEFINITION('','',#15830);
#15840=PROPERTY_DEFINITION_REPRESENTATION(#15838,#15836);

#15844=SHAPE_ASPECT('','flaskless reference occurrence',#15831,.T.);
#15846=SHAPE_DEFINING_RELATIONSHIP('','mould box reference
usage',#2030,#15844);

/* ----- pattern_equipment_available -----*/

```

```

#15900= PATTERN_EQUIPMENT_AVAILABLE('', '#42);
#15901=REPRESENTATION('quantity', (#15903, #15904, #15905, #15906, #15907, #15908,
                                #15909, #15910, #15911, #15912, #15913), #34);
#15902=PROPERTY_DEFINITION_REPRESENTATION(#15900, #15901);
#15903=DESCRIPTIVE_REPRESENTATION_ITEM('core box cavity id', 'good stuff');
#15904=DESCRIPTIVE_REPRESENTATION_ITEM('core box id', 'good stuff');
#15905=MEASURE_REPRESENTATION_ITEM('number of core
boxes', COUNT_MEASURE(1.0), #35);
#15906=MEASURE_REPRESENTATION_ITEM('number of core per
box', COUNT_MEASURE(1.0), #35);
#15907=MEASURE_REPRESENTATION_ITEM('number of cores per
casting', COUNT_MEASURE(1.0), #35);
#15908=MEASURE_REPRESENTATION_ITEM('number of
patterns', COUNT_MEASURE(1.0), #35);

#15909=DESCRIPTIVE_REPRESENTATION_ITEM('condition', 'good');
#15910=DESCRIPTIVE_REPRESENTATION_ITEM('pattern impression id', 'good stuff');
#15911=DESCRIPTIVE_REPRESENTATION_ITEM('core box material', 'epoxy');
#15912=DESCRIPTIVE_REPRESENTATION_ITEM('core per core box per
assembly', 'different core');
#15913=DESCRIPTIVE_REPRESENTATION_ITEM('pattern', 'mounted match plate');

#15921=MATERIAL_PROPERTY('', '#15922);
#15922=CHARACTERIZED_OBJECT('', '');
#15923=MATERIAL_DESIGNATION('soft wood', (#15922));
#15924=PROPERTY_DEFINITION_RELATIONSHIP('core box
material', '#15900, #15921);

#15931=MATERIAL_PROPERTY('', '#15932);
#15932=CHARACTERIZED_OBJECT('', '');
#15933=MATERIAL_DESIGNATION('soft wood', (#15932));
#15934=PROPERTY_DEFINITION_RELATIONSHIP('pattern material', '#15900, #15931);

#15936=PROPERTY_DEFINITION_RELATIONSHIP('flask size for plate
pattern', '#15900, #3356);
#15937=PROPERTY_DEFINITION_RELATIONSHIP('flask size for mounted
patterns', '#15900, #3356);
#15938=PROPERTY_DEFINITION_RELATIONSHIP('flaskless
pattern', '#15900, #15838);

#15986=PROPERTY_DEFINITION_REPRESENTATION(#15900, #15987);
#15987=IN_FACILITY_LOCATION('tool location', (#5088, #5089, #5090), #34);

/* ----- payment_and_shipping -----*/

#15940= PAYMENT_AND_SHIPPING('', '#42);
#15941=REPRESENTATION('quantity', (#15943, #15944, #15945), #34);
#15942=PROPERTY_DEFINITION_REPRESENTATION(#15940, #15941);
#15943=DESCRIPTIVE_REPRESENTATION_ITEM('customer payment', 'good stuff');
#15944=DESCRIPTIVE_REPRESENTATION_ITEM('packaging or creating', 'good stuff');
#15945=DESCRIPTIVE_REPRESENTATION_ITEM('ship_casting', 'good stuff');

/* ----- inspection_plan -----*/

```

ISO 10303-223:2008 (E)

```

#16000=(ACTION('', '#16002)
    INSPECTION_PLAN()
    PRODUCT_DEFINITION_PROCESS('')
    PROPERTY_PROCESS(''));
#16002=ACTION_METHOD('','','');
#16004=PROCESS_PRODUCT_ASSOCIATION('', '#42, #16000);
#16006=ACTION_PROPERTY('', '#16000);
#16008=ACTION_PROPERTY_REPRESENTATION('', '#16006, #16009);
#16009=REPRESENTATION('', (#16010, #16012), #34);
#16010=DESCRIPTIVE_REPRESENTATION_ITEM('test name', 'weld');
#16012=DESCRIPTIVE_REPRESENTATION_ITEM('correction
procedures', 'dimensional');
#16013=PROCESS_PROPERTY_ASSOCIATION('', '#16000, #15200);
#16014=PROCESS_PROPERTY_ASSOCIATION('', '#16000, #15300);
/* ----- request_for_quotation -----*/

#16020=REQUEST_FOR_QUOTATION('','','');
#16022=ACTION_REQUEST_SOLUTION(#16024, #16020);
#16024=ACTION_METHOD('','','');
#16026=ACTION_PROPERTY('', '#16024);
#16028=ACTION_PROPERTY_REPRESENTATION('', '#16026, #16029);
#16029=REPRESENTATION('', (#16030, #16031, #16032, #16033, #16034, #16035), #34);
#16030=MEASURE_REPRESENTATION_ITEM('quantity breaks', COUNT_MEASURE(1.0), #35);
#16031=MEASURE_REPRESENTATION_ITEM('quantity per
release', COUNT_MEASURE(1.0), #35);
#16032=MEASURE_REPRESENTATION_ITEM('quantity per
month', COUNT_MEASURE(1.0), #35);
#16033=MEASURE_REPRESENTATION_ITEM('quantity per
year', COUNT_MEASURE(1.0), #35);
#16034=MEASURE_REPRESENTATION_ITEM('total quantity per
order', COUNT_MEASURE(1.0), #35);
#16035=DESCRIPTIVE_REPRESENTATION_ITEM('type of quotation', '');

#16036=APPLIED_DATE_ASSIGNMENT(#16037, #16038, (#16020, #16060));
#16037=DATE(2006);
#16038=DATE_ROLE('delivery date');
#16039=APPLIED_DATE_ASSIGNMENT(#16037, #16041, (#16020));
#16041=DATE_ROLE('reply due date');
#16042=APPLIED_DATE_ASSIGNMENT(#16037, #16044, (#16020));
#16044=DATE_ROLE('request date');
#16045=APPLIED_DATE_ASSIGNMENT(#16037, #16047, (#16020));
#16047=DATE_ROLE('order completion');

/* ----- first_article -----*/

#16060=FIRST_ARTICLE('','','');
#16062=ACTION_REQUEST_SOLUTION(#16064, #16060);
#16064=ACTION_METHOD('','','');
#16066=ACTION_PROPERTY('', '#16064);
#16068=ACTION_PROPERTY_REPRESENTATION('', '#16066, #16069);

```



```

#16069=REPRESENTATION('', (#16070, #16071), #34);
#16070=DESCRIPTIVE_REPRESENTATION_ITEM('cavity impression id', '');
#16071=MEASURE_REPRESENTATION_ITEM('number of
pieces', COUNT_MEASURE(1.0), #35);
#16072=APPLIED_ACTION_REQUEST_ASSIGNMENT(#16060, (#16080, #17081, #17100, #18042)
);

#16073=OBJECT_ROLE('customer approval', '');
#16074=ROLE_ASSOCIATION(#16073, #16075);
#16075=APPLIED_APPROVAL_ASSIGNMENT(#1534, (#16060));

/* ----- tolerance_requirement -----*/

#16080=TOLERANCE_REQUIREMENT('', '', #42);
#16081=REPRESENTATION('quantity', (#16083, #16084, #16085, #16086, #16087), #34);
#16082=PROPERTY_DEFINITION_REPRESENTATION(#16080, #16081);
#16083=DESCRIPTIVE_REPRESENTATION_ITEM('dimension method', 'commercial
practice');
#16084=DESCRIPTIVE_REPRESENTATION_ITEM('gaging fixtures available', 'TRUE');
#16085=DESCRIPTIVE_REPRESENTATION_ITEM('tolerance based one', 'mismatch');

#16086=DESCRIPTIVE_REPRESENTATION_ITEM('new tooling required', 'foundry tool
shop');
#16087=DESCRIPTIVE_REPRESENTATION_ITEM('specific comments', ' ');

#16088=APPLIED_DOCUMENT_REFERENCE(#17002, '', (#16080));
#16089=APPLIED_DOCUMENT_REFERENCE(#17009, '', (#16080));
#16090=APPLIED_DOCUMENT_REFERENCE(#18261, '', (#16080));

#17002=DOCUMENT('', '', '', #17004);
#17004=DOCUMENT_TYPE('customer casting requirements');

#17006=PROPERTY_DEFINITION_RELATIONSHIP('', '', #16080, #17034);
#17007=PROPERTY_DEFINITION_RELATIONSHIP('', '', #16080, #17050);
#17008=PROPERTY_DEFINITION_RELATIONSHIP('', '', #16080, #78);

#17009=DOCUMENT('', '', '', #17010);
#17010=DOCUMENT_TYPE('dimension specification');

/* ----- mismatch_tolerance -----*/

#17034=MISMATCH_TOLERANCE('', '', #42);
#17036=REPRESENTATION('quantity', (#17040), #34);
#17038=PROPERTY_DEFINITION_REPRESENTATION(#17034, #17036);
#17040=MEASURE_REPRESENTATION_ITEM('maximum amount', COUNT_MEASURE(1.0), #35);

#17042=PROPERTY_DEFINITION('', '', #42);
#17044=PARTING_SURFACE('', (#600, #3613, #3614, #3616), #34);
#17046=PROPERTY_DEFINITION_REPRESENTATION(#17042, #17044);
#17047=PROPERTY_DEFINITION_RELATIONSHIP('', '', #17042, #17034);

#17048=REPRESENTATION_RELATIONSHIP('', '', #17044, #79);

```

```

/* ----- checking_aid_tool -----*/

#17050=CHECKING_AID_TOOL('', '#42);
#17052=REPRESENTATION('quantity', (#17056, #17058, #17060, #17062), #34);
#17054=PROPERTY_DEFINITION_REPRESENTATION(#17050, #17052);
#17056=DESCRIPTIVE_REPRESENTATION_ITEM('aid id', '');
#17058=DESCRIPTIVE_REPRESENTATION_ITEM('aid name', '');
#17060=DESCRIPTIVE_REPRESENTATION_ITEM('aid tool', 'guage');
#17062=DESCRIPTIVE_REPRESENTATION_ITEM('revision id', '');

#17076=APPLIED_DATE_ASSIGNMENT(#17078, #17080, (#17050));
#17078=CALENDAR_DATE(2006, 26, 07);
#17080=DATE_ROLE('revision date');

/* ----- quality_acceptance_report -----*/

#17081=DOCUMENT_FILE('', 'destruction', #17102, '', '');
#17083=DOCUMENT_REPRESENTATION_TYPE('digital', #17081);
#17084=PROPERTY_DEFINITION('document property', '', #17081);
#17085=PROPERTY_DEFINITION_REPRESENTATION(#17084, #17086);

#17086=DOCUMENT_FILE_PROPERTIES('document format', (#17087), #34);
#17087=DESCRIPTIVE_REPRESENTATION_ITEM('country code', 'USA');

#17100=DOCUMENT_FILE('', 'nde', #17102, '', '');
#17102=DOCUMENT_TYPE('quality acceptance report');
#17103=DOCUMENT_REPRESENTATION_TYPE('digital', #17100);
#17104=PROPERTY_DEFINITION('document property', '', #17100);
#17105=PROPERTY_DEFINITION_REPRESENTATION(#17104, #17086);

/* ----- simulation_report -----*/

#18000=DOCUMENT_FILE('', 'simulation', #18002, '', '');
#18002=DOCUMENT_TYPE('simulation report');
#18003=DOCUMENT_REPRESENTATION_TYPE('digital', #18000);
#18004=PROPERTY_DEFINITION('document property', '', #18000);
#18005=PROPERTY_DEFINITION_REPRESENTATION(#18004, #17086);

/* ----- master_sample -----*/

#18039=PRODUCT('', 'product design', (#38));
#18040=PRODUCT_DEFINITION_FORMATION('', 'product design', #18039);
#18041=PRODUCT_DEFINITION_CONTEXT('feature definitions', #36, 'manufacturing
planning');
#18042=MASTER_SAMPLE('', 'product design', #18040, #18041);
#18043=PRODUCT_DEFINITION_SHAPE('product shape', 'shape for product', #18042);

#18044=VERSIONED_ACTION_REQUEST('', 'project order', '', '');
#18045=ACTION_DIRECTIVE('', 'project order', '', '#18044);
#18046=ACTION_METHOD('', 'project order', '', '');
#18047=DIRECTED_ACTION('project order', 'order for project', #18046, #18045);

```

```

#18048=APPLIED_ACTION_ASSIGNMENT(#18047, (#18040, #18140));
#18049=APPLIED_APPROVAL_ASSIGNMENT(#1534, (#18047));

#18056=APPLIED_DATE_ASSIGNMENT(#18057, #18058, (#18042, #30142));
#18057=CALENDAR_DATE(2006, 26, 07);
#18058=DATE_ROLE('production date');
#18060=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #18142, #18042);
#18062=DATED_EFFECTIVITY('', #18063, #18063);
#18063=CALENDAR_DATE(2006, 26, 07);
#18064=APPLIED_EFFECTIVITY_ASSIGNMENT(#18062, (#18042));
#18066=LOT_EFFECTIVITY('', '#18067');
#18067=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #35);
#18068=APPLIED_EFFECTIVITY_ASSIGNMENT(#18066, (#18042));
#18070=PROCESS_PRODUCT_ASSOCIATION('', 'process plan used', #18042, #19270);

#18086=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #18042, #7084, 1, 'raw material', #7083);

/* ----- design_part -----*/

#18139=PRODUCT('', '', '', (#38));
#18140=PRODUCT_DEFINITION_FORMATION('', '', #18139);
#18141=PRODUCT_DEFINITION_CONTEXT('feature definitions', #36, 'manufacturing
planning');
#18142=DESIGN_PART('', 'product design', #18140, #18141);
#18143=PRODUCT_DEFINITION_SHAPE('product shape', 'shape for product', #18142);

#18186=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #18142, #7084, 1, 'raw material', #7083);

/* ----- loose_piece -----*/

#18200=(CASTING_FEATURE_DEFINITION()CASTING_INSTANCED_FEATURE()
CHARACTERIZED_OBJECT('', '')LOOSE_PIECE()SAND_CAST_DESIGN_FEATURE()SHAPE_ASPEC
T('', '', #18043, .T.));
#18202=PRODUCT_DEFINITION_SHAPE('', '', #18200);
#18212=MATERIAL_DESIGNATION('wood', (#18200));
#18214=SHAPE_ASPECT('', 'loose piece shape occurrence', #18202, .T.);
#18216=SHAPE_DEFINING_RELATIONSHIP('', 'loose piece shape usage', #561, #18214);

#18256=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#1371), #34);
#18258=PROPERTY_DEFINITION('', '', #18200);
#18260=PROPERTY_DEFINITION_REPRESENTATION(#18258, #18256);

/* ----- illustration -----*/

#18261=ILLUSTRATION('', '', 'destruction', #18262);
#18262=DOCUMENT_TYPE('');

```

```

/* ----- process_plan -----*/

#19270=PROCESS_PLAN_VERSION('', '#19271, ');

#19271=ACTION_METHOD('', '', '', '');
#19272=ACTION_METHOD_RELATIONSHIP('', '#19271, #30577);
#19273=ACTION_METHOD_RELATIONSHIP('', '#19271, #30580);

#19274=MANUFACTURING_PROCESS_RELATIONSHIP('', '#30577, #19501, 1);
#19275=MANUFACTURING_PROCESS_RELATIONSHIP('', '#19501, #19505, 1);

#19282=ACTION_RELATIONSHIP('', '#19270, #30000);
#19283=APPLIED_ACTION_ASSIGNMENT(#19270, (#30150));
#19284=ACTION_PROPERTY('', '#19270);
#19286=ACTION_PROPERTY_REPRESENTATION('', '#19284, #19287);
#19287=REPRESENTATION('', (#19288, #19289), #34);
#19288=DESCRIPTIVE_REPRESENTATION_ITEM('process plan properties', 'good
stuff');
#19289=DESCRIPTIVE_REPRESENTATION_ITEM('special instruction', 'good stuff');
#19290=PROCESS_PRODUCT_ASSOCIATION('', '#42, #19270);

/* ----- process_plan_security -----*/

#19295=PROCESS_PLAN_SECURITY('classification', 'classify product version',
#19296);
#19296=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#19297=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#19295, (#19270, #30580));
#19298=CALENDAR_DATE(2003, 27, 6);
#19299=APPLIED_DATE_ASSIGNMENT(#19298, #19300, (#19295));
#19300=DATE_ROLE('classification date');
#19301=APPLIED_DATE_ASSIGNMENT(#19298, #19302, (#19295));
#19302=DATE_ROLE('declassification date');

/* ----- revision -----*/

#19333=REVISION('', '#19270, #19470);
#19334=RELATIONSHIP_CONDITION('reason for revision', (#19333), '');
#19335=ACTION_REQUEST_SOLUTION(#19471, #19336);
#19336=VERSIONED_ACTION_REQUEST('', '', '', 'revision level');
#19337=APPLIED_APPROVAL_ASSIGNMENT(#19338, (#19333));
#19338=APPROVAL(#19339, '');
#19339=APPROVAL_STATUS('approved');
#19340=APPROVAL_DATE_TIME(#1538, #19338);
#19341=APPROVAL_PERSON_ORGANIZATION(#19345, #19338, #19342);
#19342=APPROVAL_ROLE('');
#19343=ORGANIZATIONAL_ADDRESS('', '', '', '', '', '', '', '', '', '', '', '', (#1542), 'ad
dress_or_site');
#19344=ORGANIZATION('', '', 'company');
#19345=PERSON_AND_ORGANIZATION(#19346, #19344);
#19346=PERSON('', '', '', (''), (''), (''));

```

```

#19470=PROCESS_PLAN_VERSION('', '#19471, ');

#19471=ACTION_METHOD('', '#19471, ');
#19472=ACTION_METHOD_RELATIONSHIP('', '#19471, #30577);
#19473=ACTION_METHOD_RELATIONSHIP('', '#19471, #30580);

#19482=ACTION_RELATIONSHIP('', '#19470, #30000);
#19483=APPLIED_ACTION_ASSIGNMENT(#19470, (#30150));
#19484=ACTION_PROPERTY('', '#19470);
#19486=ACTION_PROPERTY_REPRESENTATION('', '#19484, #19487);
#19487=REPRESENTATION('', (#19488, #19489), #34);
#19488=DESCRIPTIVE_REPRESENTATION_ITEM('process plan properties', 'good
stuff');
#19489=DESCRIPTIVE_REPRESENTATION_ITEM('special instruction', 'good stuff');
#19490=PROCESS_PRODUCT_ASSOCIATION('', '#42, #19470);

#19491=ALTERNATE_PLAN_RELATIONSHIP('', '#19270, #19470);
#19492=ID_ATTRIBUTE('alternate process plan', #19470);

#19500=CONTINUOUS_PROCESS_RELATIONSHIP('', 'serial', #19505, #19501, 1);

#19501=CASTING_EQUIPMENT_PROCESS('', '#19501, ');
#19502=ACTION_METHOD_RELATIONSHIP('', '#19501, #30540);
#19503=SINGLE_ACTIVITY_RELATIONSHIP('', '#19501, #30530, 1);

#19505=NON_CASTING_EQUIPMENT_PROCESS('', '#19505, ');
#19506=ACTION_METHOD_RELATIONSHIP('', '#19505, #30540);
#19507=SINGLE_ACTIVITY_RELATIONSHIP('', '#19505, #30530, 1);
/* ----- activity_execution_result -----*/

#30000=(ACTION('', '#15710)
    ACTIVITY_EXECUTION_RESULT()
    EXECUTED_ACTION()
    PROPERTY_PROCESS(''));
#30002=ACTION_METHOD('', '#30000, ');
#30004=ACTION_PROPERTY('', '#30000);
#30006=ACTION_PROPERTY_REPRESENTATION('', '#30004, #30018);
#30008=PROCESS_PROPERTY_ASSOCIATION('', 'validation part
shape', #30000, #30060);

/* ----- composition_element_record -----*/

#30018=COMPOSITION_ELEMENT_RECORD('', (#30020), #34);
#30020=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
REPRESENTATION_ITEM('actual_composition amount'));

#30022=APPLIED_DATE_AND_TIME_ASSIGNMENT(#30042, #30034, (#30018, #30158, #30198, #
30224));

```

ISO 10303-223:2008 (E)

```
#30034=DATE_TIME_ROLE('production date');
#30036=DATE(2006);
#30038=LOCAL_TIME(12,12,12.0,#30040);
#30040=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.EXACT.);
#30042=DATE_AND_TIME(#30036,#30038);
#30044=PROPERTY_DEFINITION_REPRESENTATION(#30050,#30018);

/* ----- composition_element -----*/

#30049=CHARACTERIZED_OBJECT('','');
#30050=COMPOSITION_ELEMENT('','',#30049);
#30051=REPRESENTATION('',( #30053,#30054),#34);
#30052=PROPERTY_DEFINITION_REPRESENTATION(#30050,#30051);
#30053=DESCRIPTIVE_REPRESENTATION_ITEM('element','good stuff');
#30054=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
  QUALIFIED_REPRESENTATION_ITEM((#1084,#1085,#1086))
  REPRESENTATION_ITEM('amount'));

/* ----- sampled_set -----*/

#30060=SAMPLED_SET('','',#30142);
#30061=REPRESENTATION('',( #30064),#34);
#30062=PROPERTY_DEFINITION_REPRESENTATION(#30060,#30061);

#30064=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
  REPRESENTATION_ITEM('size'));

#30068=APPLIED_EFFECTIVITY_ASSIGNMENT(#30070,(#30060));
#30070=DATED_EFFECTIVITY('',#30076,#30076);
#30076=DATE(2006);

#30078=APPLIED_EFFECTIVITY_ASSIGNMENT(#30080,(#30060));
#30080=LOT_EFFECTIVITY('','',#30082);
#30082=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0),#35);

#30088=APPLIED_DOCUMENT_REFERENCE(#30090,'',( #30060));

#30090=DOCUMENT_FILE('','',',',#30092,'','');
#30092=DOCUMENT_TYPE('production quality report');

#30091=DOCUMENT_REPRESENTATION_TYPE('digital',#30090);
#30094=PROPERTY_DEFINITION('document property','',#30090);
#30095=PROPERTY_DEFINITION_REPRESENTATION(#30094,#30096);

#30096=DOCUMENT_FILE_PROPERTIES('document format',( #30097),#34);
#30097=DESCRIPTIVE_REPRESENTATION_ITEM('country code','USA');

/* ----- master sample -----*/
```

```

#30139=PRODUCT('', '', '', (#38));
#30140=PRODUCT_DEFINITION_FORMATION('', '', #30139);
#30141=PRODUCT_DEFINITION_CONTEXT('feature      definitions', #36, 'manufacturing
planning');
#30142=MASTER_SAMPLE('', 'product design', #30140, #30141);
#30143=PRODUCT_DEFINITION_SHAPE('product shape', 'shape for product', #30142);
#30144=PRODUCT_DEFINITION_RELATIONSHIP('', '', '', #18142, #30142);
#30146=APPLIED_EFFECTIVITY_ASSIGNMENT(#30080, (#30142));
#30147=APPLIED_EFFECTIVITY_ASSIGNMENT(#30070, (#30142));

#30186=MAKE_FROM_USAGE_OPTION('1', 'tooling      material', 'raw
material', #30142, #7084, 1, 'raw material', #7083);

/* ----- material_usage -----*/

#30150=MATERIAL_USAGE('', '', #30242);
#30151=REPRESENTATION('', (#30154), #34);
#30152=PROPERTY_DEFINITION_REPRESENTATION(#30150, #30151);
#30154=(LENGTH_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
      REPRESENTATION_ITEM('amount used'));

/* ----- material_usage_record -----*/

#30158=MATERIAL_USAGE_RECORD('', (#30160), #34);
#30160=(LENGTH_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
      REPRESENTATION_ITEM('actual substance usage amount'));

#30164=PROPERTY_DEFINITION_REPRESENTATION(#30150, #30158);

/* ----- equipment_usage -----*/

#30170=EQUIPMENT_USAGE('', '', #30172, (#30174), (#3430));
#30172=RESOURCE_REQUIREMENT_TYPE('', '');
#30174=ACTION_METHOD('', '', '', '');
#30176=RESOURCE_PROPERTY('', '', #30170);
#30178=RESOURCE_PROPERTY_REPRESENTATION('', '', #30176, #30180);
#30180=REPRESENTATION('', (#30182), #34);

#30182=(LENGTH_MEASURE_WITH_UNIT()
      MEASURE_REPRESENTATION_ITEM()
      MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
      REPRESENTATION_ITEM('amount used'));

/* ----- equipment_setting_record -----*/

```

ISO 10303-223:2008 (E)

```

#30198=EQUIPMENT_SETTING_RECORD('', (#30220), #34);
#30220=DESCRIPTIVE_REPRESENTATION_ITEM('actual_parameter', 'good stuff');
#30222=RESOURCE_PROPERTY_REPRESENTATION('', '', #30176, #30198);

/* ----- process_parameter_record -----*/

#30224=PROCESS_PARAMETER_RECORD('', (#30225, #30226), #34);
#30225=DESCRIPTIVE_REPRESENTATION_ITEM('actual_parameter', 'good stuff');
#30226=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
        REPRESENTATION_ITEM('planned_parameter'));

/* ----- material -----*/

#30239=PRODUCT('', '', '', (#38));
#30240=PRODUCT_DEFINITION_FORMATION('', '', #30239);
#30241=PRODUCT_DEFINITION_CONTEXT('feature definitions', #36, 'manufacturing
planning');
#30242=PRODUCT_DEFINITION('5 X 3', 'Steel bar stock', #30240, #30241);
#30243=MATERIAL_DESIGNATION('Steel Alloy 8640', (#30242));

/* ----- shape_dimension -----*/

#30300=SHAPE_DIMENSION('', '', #30242);
#30302=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30300, #6702);

/* ----- other_inspection -----*/

#30310=OTHER_INSPECTION('', '', #30541);
#30311=REPRESENTATION('', (#30314, #30315), #34);
#30312=PROPERTY_DEFINITION_REPRESENTATION(#30310, #30311);
#30314=DESCRIPTIVE_REPRESENTATION_ITEM('inspection_result', 'good stuff');
#30315=DESCRIPTIVE_REPRESENTATION_ITEM('result', 'good stuff');
#30316=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30310, #15200);
#30317=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30310, #17050);
#30318=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30310, #30060);
#30319=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30310, #18202);

/* ----- tolerance_inspection -----*/

#30320=TOLERANCE_INSPECTION('', '', #30242);
#30321=REPRESENTATION('', (#30324, #30325), #34);
#30322=PROPERTY_DEFINITION_REPRESENTATION(#30320, #30321);
#30324=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
        REPRESENTATION_ITEM('tolerance_result'));
#30325=DESCRIPTIVE_REPRESENTATION_ITEM('result', 'good stuff');
#30326=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30320, #16080);
#30327=PROPERTY_DEFINITION_RELATIONSHIP('', '', #30320, #17050);

```



```

#30328=PROPERTY_DEFINITION_RELATIONSHIP('', '#30320', '#30060');
#30329=PROPERTY_DEFINITION_RELATIONSHIP('', '#30320', '#18202');

/* ----- compositon_inspection -----*/

#30330=COMPOSITION_INSPECTION('', '#30242');
#30331=REPRESENTATION('', (#30335), #34);
#30332=PROPERTY_DEFINITION_REPRESENTATION(#30330, #30331);

#30335=DESCRIPTIVE_REPRESENTATION_ITEM('result', 'good stuff');
#30336=PROPERTY_DEFINITION_RELATIONSHIP('', '#30330', '#13048');
#30337=PROPERTY_DEFINITION_RELATIONSHIP('', '#30330', '#17050');
#30338=PROPERTY_DEFINITION_RELATIONSHIP('', '#30330', '#30060');
#30339=PROPERTY_DEFINITION_RELATIONSHIP('', '#30330', '#18202');

/* ----- property_inspection -----*/

#30340=PROPERTY_INSPECTION('', '#30242');
#30341=REPRESENTATION('', (#30343, #30344, #30345), #34);
#30342=PROPERTY_DEFINITION_REPRESENTATION(#30340, #30341);

#30343=DESCRIPTIVE_REPRESENTATION_ITEM('result', 'good stuff');
#30344=DESCRIPTIVE_REPRESENTATION_ITEM('property result', 'good stuff');
#30345=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
  REPRESENTATION_ITEM('requirements'));

#30347=PROPERTY_DEFINITION_RELATIONSHIP('', '#30340', '#17050');
#30348=PROPERTY_DEFINITION_RELATIONSHIP('', '#30340', '#30060');
#30349=PROPERTY_DEFINITION_RELATIONSHIP('', '#30340', '#18202');

/* ----- core_assembly -----*/

#30400=(ACTION_METHOD('', '#30400')
  ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#30411))
  CASTING_METHOD()
  CORE_ASSEMBLY());
#30404=ACTION_PROPERTY('', '#30400');
#30406=ACTION_PROPERTY_REPRESENTATION('', '#30404', #30408);
#30408=REPRESENTATION('', (#30410), #34);
#30410=DESCRIPTIVE_REPRESENTATION_ITEM('hard form', 'good stuff');

#30411=DOCUMENT_FILE('', '#30412', '');
#30412=DOCUMENT_TYPE('proof report');

#30413=DOCUMENT_REPRESENTATION_TYPE('digital', #30411);
#30414=PROPERTY_DEFINITION('document property', '#30411');
#30415=PROPERTY_DEFINITION_REPRESENTATION(#30414, #30416);

#30416=DOCUMENT_FILE_PROPERTIES('document format', (#30417), #34);
#30417=DESCRIPTIVE_REPRESENTATION_ITEM('country code', 'USA');

```

ISO 10303-223:2008 (E)

```
/* ----- tool_verification -----*/

#30420=TOOL_VERIFICATION('', '', '', '');
#30421=ACTION_METHOD_RELATIONSHIP('', '', #30420, #30400);

#30422=APPLIED_ACTION_METHOD_ASSIGNMENT(#30420, #30424, (#3800, #18142, #30300, #1
6080));
#30424=ACTION_METHOD_ROLE('', '');

/* ----- allowed_time -----*/

#30450=ALLOWED_TIME('', '', #15710);
#30456=ACTION_PROPERTY_REPRESENTATION('', '', #30450, #30458);
#30458=REPRESENTATION('', (#30459, #30460, #30461), #34);
#30459=DESCRIPTIVE_REPRESENTATION_ITEM('allowed time source', 'good stuff');

#30460=(MEASURE_REPRESENTATION_ITEM(
    MEASURE_WITH_UNIT(TIME_MEASURE(2.000000E+001), #13)
    REPRESENTATION_ITEM('allowance factor')
    TIME_MEASURE_WITH_UNIT());
#30461=(MEASURE_REPRESENTATION_ITEM(
    MEASURE_WITH_UNIT(TIME_MEASURE(2.000000E+001), #13)
    REPRESENTATION_ITEM('standard time')
    TIME_MEASURE_WITH_UNIT());

/* ----- alternate_activity-----*/

#30470=ANCILLARY_ACTIVITY('', '', '', '');
#30472=ALTERNATE_ACTION_METHOD_RELATIONSHIP('', '', #30470, #15710);

#30474=ACTION_PROPERTY('', '', #30470);
#30475=ACTION_PROPERTY_REPRESENTATION('', '', #30474, #30476);
#30476=REPRESENTATION('', (#30477), #34);
#30477=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);

/* ----- externally_defined_schema-----*/

#30480=EXTERNALLY_DEFINED_SCHEMA(IDENTIFIER('externally          defined
schema'), #30482);
#30482=KNOWN_SOURCE(IDENTIFIER(''), 'ISO 10303 part');
#30488=APPLIED_DOCUMENT_REFERENCE(#30490, '', (#30482));

#30490=DOCUMENT('', '', 'externally defined schema', #30492);
#30492=DOCUMENT_TYPE('quality assurance document');

/* ----- ancillary_activity-----*/

#30500=ANCILLARY_ACTIVITY('', '', '', '');
#30504=ACTION_PROPERTY('', '', #30500);
#30505=ACTION_PROPERTY_REPRESENTATION('', '', #30504, #30506);
```

```

#30506=REPRESENTATION('', (#30507), #34);
#30507=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);

/* ----- casting_process -----*/

#30510=      (ACTION_METHOD('', '', '', ''))
            ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#15722, #18261))
            CASTING_ACTIVITY()
            CASTING_PROCESS()
            MANUFACTURING_ACTIVITY()
            PROCESS_PLAN_ACTIVITY());
#30514=ACTION_PROPERTY('', '', #30510);
#30515=ACTION_PROPERTY_REPRESENTATION('', '', #30514, #30516);
#30516=REPRESENTATION('', (#30517, #30518, #30519, #30520), #34);
#30517=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#30518=(LENGTH_MEASURE_WITH_UNIT()
        MEASURE_REPRESENTATION_ITEM()
        MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
        REPRESENTATION_ITEM('special instruction'));
#30519=DESCRIPTIVE_REPRESENTATION_ITEM('core making process', 'green sand');
#30520=DESCRIPTIVE_REPRESENTATION_ITEM('type of operation', 'die temperature
control');
#30522=DEFINING_ACTION_METHOD_RELATIONSHIP('', '', #30510, #30530);
#30528=APPLIED_ACTION_METHOD_ASSIGNMENT(#30510, #30529, (#17034, #5100, #1416));
#30529=ACTION_METHOD_ROLE('', '');

/* ----- equipment_setup -----*/

#30530=CASTING_EQUIPMENT_SETUP('', '', '', '');
#30534=ACTION_PROPERTY('', '', #30530);
#30535=ACTION_PROPERTY_REPRESENTATION('', '', #30534, #30536);
#30536=REPRESENTATION('', (#30537), #34);
#30537=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);

#30538=REQUIREMENT_FOR_ACTION_RESOURCE('', '', #30539, (#30530), (#9500));
#30539=RESOURCE_REQUIREMENT_TYPE('', '');

/* ----- dimensional_measurement_inspection -----*/

#30540=      (ACTION_METHOD('', '', '', ''))
            ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#30541))
            CASTING_ACTIVITY()
            DIMENSIONAL_MEASUREMENT_INSPECTION()
            INSPECTION_ACTIVITY()
            MANUFACTURING_ACTIVITY()
            PROCESS_PLAN_ACTIVITY());

#30541=DOCUMENT_FILE('', '', '', #30542, '', '');
#30542=DOCUMENT_TYPE('product quality report');

```

ISO 10303-223:2008 (E)

```
#30543=DOCUMENT_REPRESENTATION_TYPE('digital',#30541);
#30544=PROPERTY_DEFINITION('', '#30541');
#30545=PROPERTY_DEFINITION_REPRESENTATION(#30544,#30546);

#30546=REPRESENTATION('', (#30547,#30548,#30549),#34);
#30547=DESCRIPTIVE_REPRESENTATION_ITEM('action required','USA');
#30548=MEASURE_REPRESENTATION_ITEM('number failed',COUNT_MEASURE(1.0),#35);
#30549=DESCRIPTIVE_REPRESENTATION_ITEM('list of results','USA');

#30562=ACTION_PROPERTY('', '#30540');
#30563=ACTION_PROPERTY_REPRESENTATION('', '#30562,#30564');
#30564=REPRESENTATION('', (#30565,#30566,#30567,#30568,#30569),#34);
#30565=MEASURE_REPRESENTATION_ITEM('activity number',COUNT_MEASURE(1.0),#35);
#30566=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000),#15)
  REPRESENTATION_ITEM('special instruction'));
#30567=MEASURE_REPRESENTATION_ITEM('inspection
level',COUNT_MEASURE(1.0),#35);
#30568=DESCRIPTIVE_REPRESENTATION_ITEM('reason      for      inspection','tooling
refurbishment');
#30569=DESCRIPTIVE_REPRESENTATION_ITEM('type of inspection','dimensional');
#30570=APPLIED_ACTION_METHOD_ASSIGNMENT(#30540,#30571, (#30480,#42));
#30571=ACTION_METHOD_ROLE('', '#30540');

#30572=OBJECT_ROLE('foundry approval','');
#30573=ROLE_ASSOCIATION(#30572,#30574);
#30574=APPLIED_APPROVAL_ASSIGNMENT(#1534, (#30540));

#30575=ACTION_REQUEST_SOLUTION(#30540,#16060);
#30576=ACTION_METHOD_RELATIONSHIP('', '#30540,#16002);

/* ----- equipment_process -----*/

#30577=CASTING_EQUIPMENT_PROCESS('', '#30540');
#30578=ACTION_METHOD_RELATIONSHIP('', '#30577,#30540);
#30579=SINGLE_ACTIVITY_RELATIONSHIP('', '#30577,#30530,1);

/* ----- heat_treat -----*/

#30580=HEAT_TREAT('', '#30580);
#30581=APPLIED_ACTION_METHOD_ASSIGNMENT(#30580,#30582, (#15740));
#30582=ACTION_METHOD_ROLE('', '#30580);
#30584=ACTION_PROPERTY('', '#30580);
#30585=ACTION_PROPERTY_REPRESENTATION('', '#30584,#30586);
#30586=REPRESENTATION('', (#30587,#30588),#34);
#30587=MEASURE_REPRESENTATION_ITEM('activity number',COUNT_MEASURE(1.0),#35);
#30588=DESCRIPTIVE_REPRESENTATION_ITEM('operation description','good stuff');

/* ----- non_permanent_moulding_process -----*/

#30590=NON_PERMANENT_MOULDING_PROCESS('', 'shell', '#30590');
```

```

#30591=ACTION_PROPERTY('', '#30590);
#30592=ACTION_PROPERTY_REPRESENTATION('', '#30591, #30593);
#30593=REPRESENTATION('', (#30594, #30595, #30596, #30597), #34);
#30594=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#30595=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
  REPRESENTATION_ITEM('special instruction'));
#30596=MEASURE_REPRESENTATION_ITEM('number of
impressions', COUNT_MEASURE(1.0), #35);
#30597=DESCRIPTIVE_REPRESENTATION_ITEM('pattern form', 'loose pattern with
follow block');
#30598=APPLIED_ACTION_METHOD_ASSIGNMENT(#30590, #30599, (#3356, #5006));
#30599=ACTION_METHOD_ROLE('', '');

/* ----- tooling_process -----*/

#30600=TOOLING_PROCESS('', '#30600);

#30601=ACTION_PROPERTY('', '#30600);
#30602=ACTION_PROPERTY_REPRESENTATION('', '#30601, #30603);
#30603=REPRESENTATION('', (#30604, #30605, #30606, #30607, #30608), #34);
#30604=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);
#30605=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
  REPRESENTATION_ITEM('special instruction'));
#30606=MEASURE_REPRESENTATION_ITEM('number of
impressions', COUNT_MEASURE(1.0), #35);
#30607=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0620000000000000), #15)
  REPRESENTATION_ITEM('shrinkage factor'));
#30608=DESCRIPTIVE_REPRESENTATION_ITEM('process method', 'non permanent
moulding process');

#30610=ACTION_METHOD_RELATIONSHIP('', '#30600, #30420);
#30612=APPLIED_ACTION_METHOD_ASSIGNMENT(#30600, #30613, (#3706, #4006, #15400, #18
142));
#30613=ACTION_METHOD_ROLE('', '');

/* ----- simulation_process and metalcaster_simulation -----*/

#30700= (ACTION_METHOD('', '#30700)
ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#30707, #30710))
CASTING_ACTIVITY()
MANUFACTURING_ACTIVITY()
METALCASTER_SIMULATION()
PROCESS_PLAN_ACTIVITY()

```

## ISO 10303-223:2008 (E)

```
SIMULATION_PROCESS();

#30701=ACTION_PROPERTY('', '#30700');
#30702=ACTION_PROPERTY_REPRESENTATION('', '#30701, #30703');
#30703=REPRESENTATION('', (#30704), #34);
#30704=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);

#30707=DOCUMENT('', '#30708');
#30708=DOCUMENT_TYPE('simulation exception report');

#30710=DOCUMENT('', '#30712');
#30712=DOCUMENT_TYPE('simulation tooling report');

#30714=APPLIED_ACTION_METHOD_ASSIGNMENT(#30700, #30716, (#15400, #15900, #4006, #7000));
#30716=ACTION_METHOD_ROLE('', '');

/* ----- customer_simulation -----*/

#30720=      (ACTION_METHOD('', '#30720')
              ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS((#30727, #30730))
              CASTING_ACTIVITY()
              CUSTOMER_SIMULATION()
              MANUFACTURING_ACTIVITY()
              PROCESS_PLAN_ACTIVITY()
              SIMULATION_PROCESS());

#30721=ACTION_PROPERTY('', '#30720');
#30722=ACTION_PROPERTY_REPRESENTATION('', '#30721, #30723');
#30723=REPRESENTATION('', (#30724), #34);
#30724=MEASURE_REPRESENTATION_ITEM('activity number', COUNT_MEASURE(1.0), #35);

#30727=DOCUMENT('', '#30728');
#30728=DOCUMENT_TYPE('customer tool design report');

#30730=DOCUMENT('', '#30732');
#30732=DOCUMENT_TYPE('customer design deficiency report');

#30734=APPLIED_ACTION_METHOD_ASSIGNMENT(#30720, #30736, (#15400, #7000));
#30736=ACTION_METHOD_ROLE('', '');

/* ----- simulation_run -----*/

#30800=SIMULATION_RUN('', '#30700, #30804');
#30802=ID_ATTRIBUTE('', #30800);
#30804=ACTION_DIRECTIVE('', '#31100');

#30810=SIMULATION_RUN('', '#30720, #30814');
#30812=ID_ATTRIBUTE('', #30810);
#30814=ACTION_DIRECTIVE('', '#31100');
```

```

/* ----- simulation_input -----*/

#30900=SIMULATION_INPUT('', '#30902', (#30800, #30810, #31300, #31302, #31708), (#31000));
#30902=RESOURCE_REQUIREMENT_TYPE('', '');

#30904=APPLIED_DOCUMENT_REFERENCE(#30905, '', (#30900, #31300, #31302));
#30905=DOCUMENT_FILE('', 'simulation document', #30906, '', '');
#30906=DOCUMENT_TYPE('');
#30907=DOCUMENT_REPRESENTATION_TYPE('digital', #30905);

#30908=RESOURCE_PROPERTY('', '#30900');
#30910=RESOURCE_PROPERTY_REPRESENTATION('', '#30908', #30912);
#30912=REPRESENTATION('', (#30914), #34);
#30914=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('input parameter'));
#30916=RESOURCE_PROPERTY_REPRESENTATION('', '#30908', #31400);

/* ----- simulation_input_region -----*/

#31000=SIMULATION_INPUT_REGION('', '#31004', (#30800, #31508, #31004, #31708, #31114, #31608), #31002);
#31002=ACTION_RESOURCE_TYPE('');
#31004=ACTION_METHOD('', 'input', 'region');

#31010=APPLIED_ACTION_METHOD_ASSIGNMENT(#31004, #31012, (#42, #77, #31014, #31022));
#31012=ACTION_METHOD_ROLE('', '#31004');

#31014=PROPERTY_DEFINITION('', '#42');
#31016=PROPERTY_DEFINITION_REPRESENTATION(#31014, #31018);

#31018=REPRESENTATION('', (#31020), #34);
#31020=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
  REPRESENTATION_ITEM('region property'));

#31022=PROPERTY_DEFINITION('', '#42');
#31024=PROPERTY_DEFINITION_REPRESENTATION(#31022, #31400);

/* ----- simulation_output -----*/

#31100=SIMULATION_OUTPUT('', '#31102', 'output', 'region');
#31102=ACTION_REQUEST_SOLUTION(#31150, #31100);

/* ----- simulation_output_region -----*/

#31110=SIMULATION_OUTPUT_REGION('', '#31114', #31114);
#31114=ACTION_METHOD('', 'output', 'region');

```

ISO 10303-223:2008 (E)

```
#31116=APPLIED_ACTION_METHOD_ASSIGNMENT(#31114,#31118,(#77));
#31118=ACTION_METHOD_ROLE('','');
#31119=ACTION_PROPERTY_RELATIONSHIP('','',#31110,#31120);

/* ----- simulation_unit_state -----*/

#31120=SIMULATION_UNIT_STATE('','',#31124);
#31124=ACTION_METHOD('','','');

#31126=APPLIED_TIME_ASSIGNMENT(#31128,#31130,(#31120));
#31128=LOCAL_TIME(12,6,12.2,#31168);
#31130=TIME_ROLE('evaluation time');

#31132=ACTION_PROPERTY_REPRESENTATION('','',#31120,#31141);

/* ----- simulation_unit -----*/

#31133=SIMULATION_UNIT('','',#31134);
#31134=ACTION_METHOD('','','');
#31135=APPLIED_ACTION_METHOD_ASSIGNMENT(#31134,#31136,(#77,#31154));
#31136=ACTION_METHOD_ROLE('','');
#31137=ACTION_PROPERTY_RELATIONSHIP('','',#31133,#31120);
#31138=ACTION_PROPERTY_RELATIONSHIP('','',#31133,#31110);
/* ----- simulated_property -----*/

#31141=SIMULATED_PROPERTY('',( #31142),#34);
#31142=MEASURE_REPRESENTATION_ITEM('property
value',PARAMETER_VALUE(1.0),#35);

/* ----- defect_prediction -----*/

#31150=DEFECT_PREDICTION('','','');
#31151=APPLIED_ACTION_METHOD_ASSIGNMENT(#31150,#31152,(#77,#31154));
#31152=ACTION_METHOD_ROLE('','');

#31154=PROPERTY_DEFINITION('','',#42);
#31156=PROPERTY_DEFINITION_REPRESENTATION(#31154,#31158);
#31158=LOCATION_SHAPE_REPRESENTATION('location',(#595),#34);

/* ----- simulation_result-----*/

#31160=SIMULATION_RESULT(#31162,#31100);
#31162=ACTION_METHOD('','','');

#31163=APPLIED_TIME_ASSIGNMENT(#31164,#31166,(#31160));
#31164=LOCAL_TIME(12,6,12.2,#31168);
#31166=TIME_ROLE('begin time');
#31168=COORDINATED_UNIVERSAL_TIME_OFFSET(0,0,.EXACT.);

#31173=APPLIED_TIME_ASSIGNMENT(#31174,#31176,(#31160));
#31174=LOCAL_TIME(12,6,12.2,#31168);
#31176=TIME_ROLE('end time');
```



```

#31180=ACTION_METHOD_RELATIONSHIP('', '#31162', '#31114');
#31182=APPLIED_DOCUMENT_REFERENCE(#18261, '#31160');

/* ----- simulation_software-----*/

#31200=SIMULATION_SOFTWARE('', '#30800', '#30810', '#31210');
#31210=ACTION_RESOURCE_TYPE('');

#31211=APPLIED_DATE_ASSIGNMENT(#31212, '#31214', '#31200');
#31212=DATE(2006);
#31214=DATE_ROLE('release date');

/* ----- condition_or_assumption-----*/

#31300=CONDITION_OR_ASSUMPTION('condition', '', '', '');
#31302=CONDITION_OR_ASSUMPTION('assumption', '', '', '');

#31310=APPLIED_ACTION_METHOD_ASSIGNMENT(#31300, '#31312', (#77, '#31314', '#31322));
#31311=APPLIED_ACTION_METHOD_ASSIGNMENT(#31302, '#31312', (#77, '#31314', '#31322));
#31312=ACTION_METHOD_ROLE('', '');

#31314=PROPERTY_DEFINITION('', '#42');
#31316=PROPERTY_DEFINITION_REPRESENTATION(#31314, '#31318');

#31318=REPRESENTATION('', '#31320', '#34');
#31320=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), '#15)
  REPRESENTATION_ITEM('condition value'));

#31322=PROPERTY_DEFINITION('', '#42');
#31324=PROPERTY_DEFINITION_REPRESENTATION(#31322, '#31400);

/* ----- property_relationship-----*/

#31400=PROPERTY_RELATIONSHIP('input parameter', (#31402, '#31404), '#34);
#31402=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), '#15)
  REPRESENTATION_ITEM('x property name'));
#31404=(LENGTH_MEASURE_WITH_UNIT()
  MEASURE_REPRESENTATION_ITEM()
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), '#15)
  REPRESENTATION_ITEM('y property name'));
#31405=REPRESENTATION_RELATIONSHIP('', '#31400', '#31406);

/* ----- data curve-----*/

#31406=DATA_CURVE('relationship curve', (#44, #45, #31408, #31410, #31412), '#34);

```

ISO 10303-223:2008 (E)

```
#31408=(LENGTH_MEASURE_WITH_UNIT()  
  MEASURE_REPRESENTATION_ITEM()  
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)  
  REPRESENTATION_ITEM('x unit'));  
  
#31410=(LENGTH_MEASURE_WITH_UNIT()  
  MEASURE_REPRESENTATION_ITEM()  
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)  
  REPRESENTATION_ITEM('y unit'));  
  
#31412=DESCRIPTIVE_REPRESENTATION_ITEM('interpolation method','');  
  
/* ----- boundary_condition-----*/  
  
#31508=BOUNDARY_CONDITION('','','','');  
#31510=APPLIED_ACTION_METHOD_ASSIGNMENT(#31508,#31512, (#77,#31514,#31522));  
#31512=ACTION_METHOD_ROLE('','');  
  
#31514=PROPERTY_DEFINITION('','#42);  
#31516=PROPERTY_DEFINITION_REPRESENTATION(#31514,#31518);  
  
#31518=REPRESENTATION('',( #31520), #34);  
#31520=(LENGTH_MEASURE_WITH_UNIT()  
  MEASURE_REPRESENTATION_ITEM()  
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)  
  REPRESENTATION_ITEM('condition value'));  
  
#31522=PROPERTY_DEFINITION('','#42);  
#31524=PROPERTY_DEFINITION_REPRESENTATION(#31522,#31400);  
  
/* ----- meshing_condition-----*/  
  
#31608=MESHING_CONDITION('','','','');  
#31610=APPLIED_ACTION_METHOD_ASSIGNMENT(#31608,#31612, (#77,#31614,#31632));  
#31612=ACTION_METHOD_ROLE('','');  
  
#31614=PROPERTY_DEFINITION('','#42);  
#31616=PROPERTY_DEFINITION_REPRESENTATION(#31614,#31618);  
  
#31618=REPRESENTATION('',( #31620,#31622,#31624,#31626), #34);  
  
#31620=MEASURE_REPRESENTATION_ITEM('number of elements',COUNT_MEASURE(1.0),#35);  
#31622=MEASURE_REPRESENTATION_ITEM('number of nodes',COUNT_MEASURE(1.0),#35);  
  
#31624=(LENGTH_MEASURE_WITH_UNIT()  
  MEASURE_REPRESENTATION_ITEM()  
  MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)  
  REPRESENTATION_ITEM('density of nodes'));  
  
#31626=(LENGTH_MEASURE_WITH_UNIT()  
  MEASURE_REPRESENTATION_ITEM()
```

```

MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000),#15)
REPRESENTATION_ITEM('density of elements'));

#31632=PROPERTY_DEFINITION('', '#42');
#31634=PROPERTY_DEFINITION_REPRESENTATION(#31632, #31400);

/* ----- integration_interval-----*/

#31708=INTEGRATION_INTERVAL('assumption', '', '', '');
#31710=APPLIED_ACTION_METHOD_ASSIGNMENT(#31708, #31712, (#77, #31714));
#31712=ACTION_METHOD_ROLE('', '');

#31714=PROPERTY_DEFINITION('', '#42');
#31716=PROPERTY_DEFINITION_REPRESENTATION(#31714, #31718);

#31718=REPRESENTATION('', (#31724, #31726), #34);

#31724=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
REPRESENTATION_ITEM('delta time'));

#31726=(LENGTH_MEASURE_WITH_UNIT()
MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1300000000000000), #15)
REPRESENTATION_ITEM('length of elapsed time interval'));

#31800=(CHARACTERIZED_OBJECT('', '') FEATURE_DEFINITION() INSTANCED_FEATURE()
SHAPE_ASPECT('', '#43, .T.) THREAD());
#31802=PRODUCT_DEFINITION_SHAPE('', '#31800);
#31803=CARTESIAN_POINT('', (0.0, 0.0, 120.0));
#31804=DIRECTION('', (0.0, 0.0, 1.0));
#31805=DIRECTION('', (1.0, 0.0, 0.0));
#31806=AXIS2_PLACEMENT_3D('orientation', #31803, #31804, #31805);

#31807=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #1
2)
RATIO_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('number of threads'));
#31808=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.4375), #10)
REPRESENTATION_ITEM('major diameter'));
#31809=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.43), #10)
REPRESENTATION_ITEM('minor diameter'));
#31810=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(0.433), #10)
REPRESENTATION_ITEM('pitch diameter'));
#31811=DESCRIPTIVE_REPRESENTATION_ITEM('fit class', '2A');
#31812=DESCRIPTIVE_REPRESENTATION_ITEM('form', 'UNF');
#31813=DESCRIPTIVE_REPRESENTATION_ITEM('hand', 'right');
#31814=DESCRIPTIVE_REPRESENTATION_ITEM('qualifier', '');
#31815=DESCRIPTIVE_REPRESENTATION_ITEM('thread system', 'english');
#31816=DESCRIPTIVE_REPRESENTATION_ITEM('thread side', 'internal');

```

ISO 10303-223:2008 (E)

```

#31818=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#31806, #31807, #31808, #31809, #
31810, #31811, #31812, #31813, #31814, #31815, #31816), #34);
#31819=SHAPE_DEFINITION_REPRESENTATION(#31802, #31818);

#31836=SHAPE_ASPECT('', 'partial area occurrence', #31802, .T.);
#31837=SHAPE_DEFINING_RELATIONSHIP('', 'applied shape', #31838, #31836);
#31838=SHAPE_ASPECT('featured shape', '', #43, .T.);
#31839=PROPERTY_DEFINITION('', '', #31838);
#31840=SHAPE_REPRESENTATION('applied shape', (#968, #950), #34);
#31841=SHAPE_DEFINITION_REPRESENTATION(#31839, #31840);

#31924=FEATURE_COMPONENT_DEFINITION('', '');
#31925=PRODUCT_DEFINITION_SHAPE('', '', #31924);
#31927=APPLIED_AREA('bottom condition', '', #31925, .F.);
#31931=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()
MEASURE_WITH_UNIT(LENGTH_MEASURE(27.0), #10)REPRESENTATION_ITEM('length'));
#31932=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#31931), #34);
#31933=SHAPE_DEFINITION_REPRESENTATION(#31925, #31932);

#31942=SHAPE_ASPECT('', 'partial area occurrence', #31802, .T.);
#31943=SHAPE_DEFINING_RELATIONSHIP('applied          shape', 'applied          area
usage', #31927, #31942);

/* -----          red_line_part -----*/

#32000=PRODUCT('', '', '', (#38));
#32002=PRODUCT_DEFINITION_FORMATION('', '', #32000);
#32004=PRODUCT_DEFINITION_CONTEXT('feature          definitions', #36, 'manufacturing
planning');
#32006=RED_LINED_PART('', 'product design', #32002, #32004);
#32008=PRODUCT_DEFINITION_SHAPE('product shape', 'shape for product', #32006);
#32010=APPLIED_ACTION_ASSIGNMENT(#1531, (#32002));
#32012=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#15005, (#32002));

/* -----          red_lined_part to property (as mark_up_property) -----
-----*/

#32015=ID_ATTRIBUTE('mark up property', #32019);
#32016=DESCRIPTIVE_REPRESENTATION_ITEM('note text', 'HEAT TREAT PER MIL-H-6875
TO 150,000-160,000 PSI. ');
#32017=REPRESENTATION('note text', (#32016), #34);
#32019=PROPERTY_DEFINITION('part property', 'part note', #32006);
#32020=PROPERTY_DEFINITION_REPRESENTATION(#32019, #32017);

/* -----          red_lined_part to material (as mark_up_material) -----
-----*/

#32024=MAKE_FROM_USAGE_OPTION('1', 'mark          up          material', 'product
material', #32006, #1523, 1, 'primary material', #1522);

```

```

/* ----- red_lined_part to shape_aspect (as mark_up_shape) ---
-----*/

#32042=SHAPE_ASPECT('', 'red lined part occurrence', #32008, .T.);
#32043=SHAPE_DEFINING_RELATIONSHIP('applied shape', 'red lined shape
usage', #697, #32042);

/* ----- red_lined_part to feature (as mark_up_feature) -----
-----*/

#32045=SHAPE_DEFINING_RELATIONSHIP('applied shape', 'red lined feature
usage', #2030, #32042);

/* ----- cast_part_with_rigging -----*/

#33000=PRODUCT('', '', '', (#38));
#33002=PRODUCT_DEFINITION_FORMATION('', '', #33000);
#33004=PRODUCT_DEFINITION_CONTEXT('feature definitions', #36, 'manufacturing
planning');
#33006=CAST_PART_WITH_RIGGING('', 'product design', #33002, #33004);
#33008=PRODUCT_DEFINITION_SHAPE('product shape', 'shape for product', #33006);
#33010=APPLIED_ACTION_ASSIGNMENT(#1531, (#33002));
#33012=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#15005, (#33002));

#33086=MAKE_FROM_USAGE_OPTION('1', 'tooling material', 'raw
material', #33006, #7084, 1, 'raw material', #7083);

/* ----- material_composition and composition_element ----- */

#41538=PRODUCT_CONTEXT('chemical composition', #36, 'Forging');
#41539=PRODUCT('', '', '', (#41538));
#41540=PRODUCT_DEFINITION_FORMATION('', '', #41539);
#41541=PRODUCT_DEFINITION_CONTEXT('material definition', #36, 'Forging');
#41523=PRODUCT_DEFINITION('3/8 OD X 4 5/16 LG', '4130 STEEL COND D & 4 PER
MIL-S-6758', #41540, #41541);
#41524=MATERIAL_DESIGNATION('4130 STEEL COND D & 4 PER MIL-S-6758', (#41523));

#41638=PRODUCT_CONTEXT('chemical composition element', #36, 'Forging');
#41639=PRODUCT('', 'Iron', '', (#41638));
#41640=PRODUCT_DEFINITION_FORMATION('', '', #41639);
#41641=PRODUCT_DEFINITION_CONTEXT('material composition definition', #36, '');
#41623=PRODUCT_DEFINITION('', '', #41640, #41641);
#41624=MATERIAL_DESIGNATION('', (#41623));

#41700=PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP('material composition
element', '', '', #41523, #41623,
'', (#41704), '', '');
#41702=REPRESENTATION('', (#41704), #34);
#41704=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #1
2)
RATIO_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('weight fraction percent'));

```

## ISO 10303-223:2008 (E)

```
#41838=PRODUCT_CONTEXT('chemical composition element',#36,'Forging');
#41839=PRODUCT('', 'Tin', '', (#41838));
#41840=PRODUCT_DEFINITION_FORMATION('', '', #41839);
#41841=PRODUCT_DEFINITION_CONTEXT('material composition definition',#36,'');
#41823=PRODUCT_DEFINITION('', '', #41840, #41841);
#41824=MATERIAL_DESIGNATION('', (#41823));

#41900=PRODUCT_MATERIAL_COMPOSITION_RELATIONSHIP('material composition
element', '', '', #41523, #41823,
'', (#41904), '', '');
#41902=REPRESENTATION('', (#41904), #34);
#41904=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #1
2)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('weight fraction percent'));

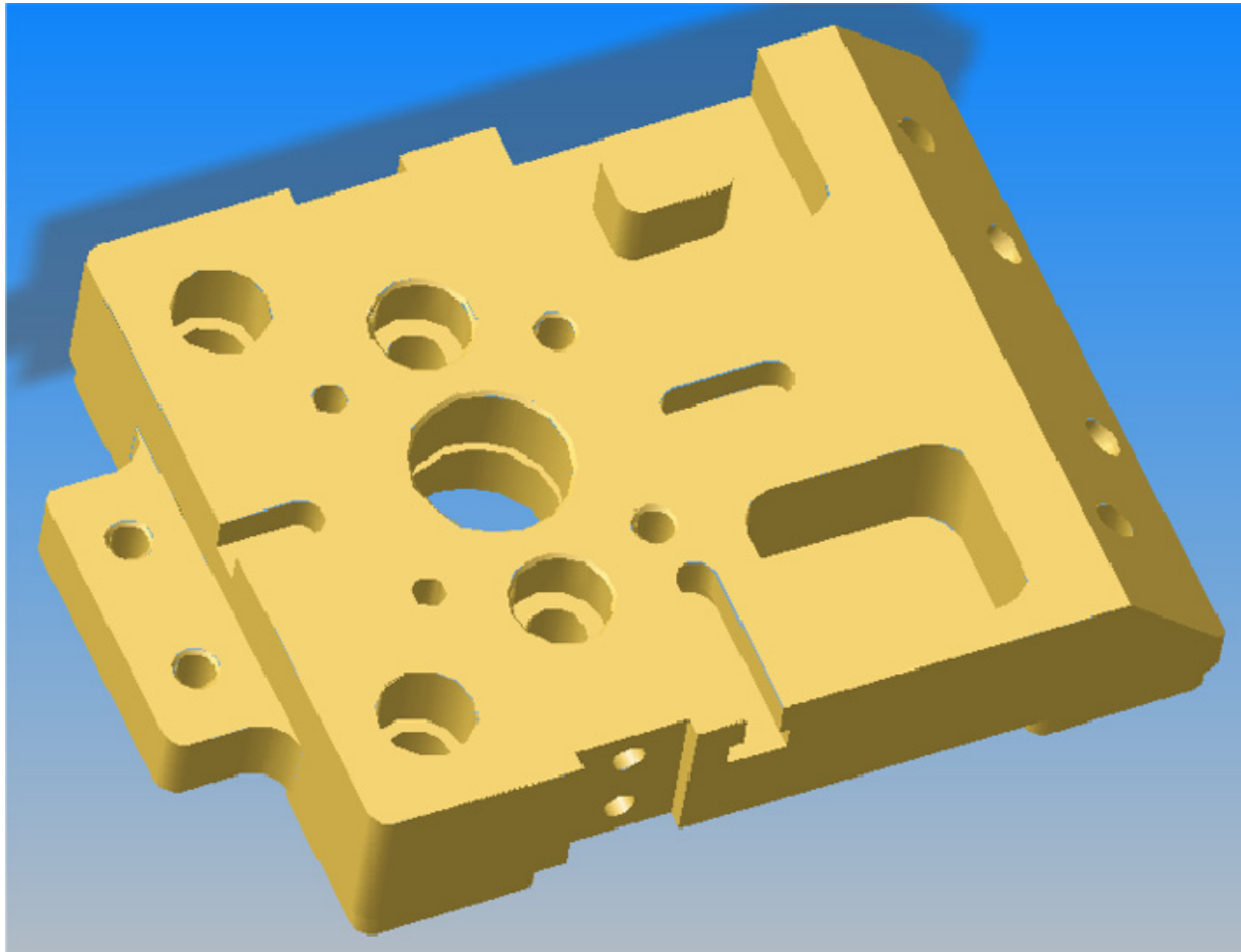
ENDSEC;
END-ISO-10303-21;
```

### J.6 Manufacturing suite test case

This test case was generated to test Part 240, Part 238, Part 224 and this Part. The part was modified by adding AP223 specific data, and removing PART 224 specific data. This part has now been generated on a CAD system and output as an Part 224 file. This Part 224 file has been tested and validated to the Part 224 schema. The data that is in common between all of the application protocols includes: Boundary-representation geometry, Features, Geometric and Dimensional Tolerances, Materials, and some product structure.

This part was successfully tested against the schema in this document.





**Figure J.2 — Manufacturing suite test case part**

```

ISO-10303-21;
HEADER;
FILE_DESCRIPTION(('AP224 File'),'2;1');
FILE_NAME('00000_ap223-101_01.224','2003-05-02T08:50:53-05:00',('DICK
TIANO'),('SCRA'),'STEPTrans 8.1.0 (Build 2003-04-02; 11:40:23 EST)','PTC
Pro/ENGINEER Version 2001','DICK TIANO');
FILE_SCHEMA(('cast_parts_schema'));
ENDSEC;
DATA;
#5=DIMENSIONAL_EXPONENTS(1.0,0.0,0.0,0.0,0.0,0.0,0.0);
#6=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#10=(LENGTH_UNIT()NAMED_UNIT(#5)SI_UNIT(.MILLI.,.METRE.));
#11=RATIO_UNIT(#6);
#14=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#15=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.017453292519943),#14)
;
#19=(CONVERSION_BASED_UNIT('DEGREE',#15)NAMED_UNIT(#6)PLANE_ANGLE_UNIT());
#23=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
    
```



```

#24=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0000100000000000),#10,'closure','maximum model space distance between geometric entities at asserted connectivities');
#29=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#24))GLOBAL_UNIT_ASSIGNED_CONTEXT((#10,#19,#23))REPRESENTATION_CONTEXT('ID1','3D'));
#30=CONTEXT_DEPENDENT_UNIT(#6,'COUNT');

#31=APPLICATION_CONTEXT('cast parts schema');
#32=APPLICATION_PROTOCOL_DEFINITION('IS','cast parts schema',2000,#31);
#33=PRODUCT_CONTEXT('Feature Based Process Planning',#31,'Mechanical');
#34=PRODUCT('00000_ap223-101_01','STEP NC SUPER MODEL TEST PART',$(#33));
#35=PRODUCT_DEFINITION_FORMATION('NA','NA',#34);
#36=PRODUCT_DEFINITION_CONTEXT('part definition',#31,'manufacturing planning');
#37=PRODUCT_DEFINITION('00000_AP223-101','product design',#35,#36);
#38=PRODUCT_DEFINITION_SHAPE('product shape','shape for product',#37);
#39=CARTESIAN_POINT('',(225.0,-40.0,0.0));
#40=VERTEX_POINT('',#39);
#41=CARTESIAN_POINT('',(5.0,-40.0,0.0));
#42=VERTEX_POINT('',#41);
#43=DIRECTION('',(-1.0,0.0,0.0));
#44=VECTOR('',#43,220.0);
#45=LINE('',#39,#44);
#46=EDGE_CURVE('',#40,#42,#45,.T.);
#47=ORIENTED_EDGE('',*,*,#46,.F.);
#48=CARTESIAN_POINT('',(225.0,0.0,0.0));
#49=VERTEX_POINT('',#48);
#50=DIRECTION('',(0.0,-1.0,0.0));
#51=VECTOR('',#50,40.0);
#52=LINE('',#48,#51);
#53=EDGE_CURVE('',#49,#40,#52,.T.);
#54=ORIENTED_EDGE('',*,*,#53,.F.);
#55=CARTESIAN_POINT('',(5.0,0.0,0.0));
#56=VERTEX_POINT('',#55);
#57=DIRECTION('',(-1.0,0.0,0.0));
#58=VECTOR('',#57,220.0);
#59=LINE('',#48,#58);
#60=EDGE_CURVE('',#49,#56,#59,.T.);
#61=ORIENTED_EDGE('',*,*,#60,.T.);
#62=DIRECTION('',(0.0,-1.0,0.0));
#63=VECTOR('',#62,40.0);
#64=LINE('',#55,#63);
#65=EDGE_CURVE('',#56,#42,#64,.T.);
#66=ORIENTED_EDGE('',*,*,#65,.T.);
#67=EDGE_LOOP('',(#47,#54,#61,#66));
#68=FACE_BOUND('',#67,.F.);
#69=CARTESIAN_POINT('',(0.0,0.0,0.0));
#70=DIRECTION('',(0.0,0.0,1.0));
#71=DIRECTION('',(1.0,0.0,0.0));
#72=AXIS2_PLACEMENT_3D('',#69,#70,#71);
#73=PLANE('',#72);
#74=ADVANCED_FACE('',(#68),#73,.F.);

```

```

#75=SHAPE_ASPECT('featured shape: face 1', $, #38, .T.);
#76=PROPERTY_DEFINITION('', $, #75);
#77=FACE_SHAPE_REPRESENTATION('', (#74), #29);
#78=PROPERTY_DEFINITION_REPRESENTATION(#76, #77);
#79=CARTESIAN_POINT('', (47.999999999999979, -20.0, 180.0));
#80=VERTEX_POINT('', #79);
#81=CARTESIAN_POINT('', (117.999999999999943, -20.0, 180.0));
#82=VERTEX_POINT('', #81);
#83=DIRECTION('', (1.0, 0.0, 0.0));
#84=VECTOR('', #83, 69.999999999999972);
#85=LINE('', #79, #84);
#86=EDGE_CURVE('', #80, #82, #85, .T.);
#87=ORIENTED_EDGE('', *, *, #86, .T.);
#88=CARTESIAN_POINT('', (117.999999999999943, 0.0, 180.0));
#89=VERTEX_POINT('', #88);
#90=DIRECTION('', (0.0, -1.0, 0.0));
#91=VECTOR('', #90, 20.0);
#92=LINE('', #88, #91);
#93=EDGE_CURVE('', #89, #82, #92, .T.);
#94=ORIENTED_EDGE('', *, *, #93, .F.);
#95=CARTESIAN_POINT('', (225.0, 0.0, 180.0));
#96=VERTEX_POINT('', #95);
#97=DIRECTION('', (-1.0, 0.0, 0.0));
#98=VECTOR('', #97, 107.0000000000000057);
#99=LINE('', #95, #98);
#100=EDGE_CURVE('', #96, #89, #99, .T.);
#101=ORIENTED_EDGE('', *, *, #100, .F.);
#102=CARTESIAN_POINT('', (225.0, -40.0, 180.0));
#103=VERTEX_POINT('', #102);
#104=DIRECTION('', (0.0, 1.0, 0.0));
#105=VECTOR('', #104, 40.0);
#106=LINE('', #102, #105);
#107=EDGE_CURVE('', #103, #96, #106, .T.);
#108=ORIENTED_EDGE('', *, *, #107, .F.);
#109=CARTESIAN_POINT('', (5.0, -40.0, 180.0));
#110=VERTEX_POINT('', #109);
#111=DIRECTION('', (1.0, 0.0, 0.0));
#112=VECTOR('', #111, 220.0);
#113=LINE('', #109, #112);
#114=EDGE_CURVE('', #110, #103, #113, .T.);
#115=ORIENTED_EDGE('', *, *, #114, .F.);
#116=CARTESIAN_POINT('', (5.0, 0.0, 180.0));
#117=VERTEX_POINT('', #116);
#118=DIRECTION('', (0.0, -1.0, 0.0));
#119=VECTOR('', #118, 40.0);
#120=LINE('', #116, #119);
#121=EDGE_CURVE('', #117, #110, #120, .T.);
#122=ORIENTED_EDGE('', *, *, #121, .F.);
#123=CARTESIAN_POINT('', (47.999999999999979, 0.0, 180.0));
#124=VERTEX_POINT('', #123);
#125=DIRECTION('', (-1.0, 0.0, 0.0));
#126=VECTOR('', #125, 42.999999999999979);
#127=LINE('', #123, #126);

```

```

#128=EDGE_CURVE('', #124, #117, #127, .T.);
#129=ORIENTED_EDGE('', *, *, #128, .F.);
#130=DIRECTION('', (0.0, -1.0, 0.0));
#131=VECTOR('', #130, 20.0);
#132=LINE('', #123, #131);
#133=EDGE_CURVE('', #124, #80, #132, .T.);
#134=ORIENTED_EDGE('', *, *, #133, .T.);
#135=EDGE_LOOP('', (#87, #94, #101, #108, #115, #122, #129, #134));
#136=FACE_BOUND('', #135, .F.);
#137=CARTESIAN_POINT('', (0.0, 0.0, 180.0));
#138=DIRECTION('', (0.0, 0.0, 1.0));
#139=DIRECTION('', (1.0, 0.0, 0.0));
#140=AXIS2_PLACEMENT_3D('', #137, #138, #139);
#141=PLANE('', #140);
#142=ADVANCED_FACE('', (#136), #141, .T.);
#143=SHAPE_ASPECT('featured shape: face 2', $, #38, .T.);
#144=PROPERTY_DEFINITION('', $, #143);
#145=FACE_SHAPE_REPRESENTATION('', (#142), #29);
#146=PROPERTY_DEFINITION_REPRESENTATION(#144, #145);
#147=CARTESIAN_POINT('', (0.0, -30.0, 106.0));
#148=VERTEX_POINT('', #147);
#149=CARTESIAN_POINT('', (0.0, -10.0, 106.0));
#150=VERTEX_POINT('', #149);
#151=DIRECTION('', (0.0, 1.0, 0.0));
#152=VECTOR('', #151, 20.0);
#153=LINE('', #147, #152);
#154=EDGE_CURVE('', #148, #150, #153, .T.);
#155=ORIENTED_EDGE('', *, *, #154, .T.);
#156=CARTESIAN_POINT('', (0.0, -10.0, 99.0));
#157=VERTEX_POINT('', #156);
#158=DIRECTION('', (0.0, 0.0, 1.0));
#159=VECTOR('', #158, 7.0);
#160=LINE('', #156, #159);
#161=EDGE_CURVE('', #157, #150, #160, .T.);
#162=ORIENTED_EDGE('', *, *, #161, .F.);
#163=CARTESIAN_POINT('', (0.0, 0.0, 99.0));
#164=VERTEX_POINT('', #163);
#165=DIRECTION('', (0.0, -1.0, 0.0));
#166=VECTOR('', #165, 10.0);
#167=LINE('', #163, #166);
#168=EDGE_CURVE('', #164, #157, #167, .T.);
#169=ORIENTED_EDGE('', *, *, #168, .F.);
#170=CARTESIAN_POINT('', (0.0, 0.0, 175.0));
#171=VERTEX_POINT('', #170);
#172=DIRECTION('', (0.0, 0.0, 1.0));
#173=VECTOR('', #172, 76.0);
#174=LINE('', #163, #173);
#175=EDGE_CURVE('', #164, #171, #174, .T.);
#176=ORIENTED_EDGE('', *, *, #175, .T.);
#177=CARTESIAN_POINT('', (0.0, -40.0, 175.0));
#178=VERTEX_POINT('', #177);
#179=DIRECTION('', (0.0, -1.0, 0.0));
#180=VECTOR('', #179, 40.0);

```

ISO 10303-223:2008 (E)

```

#181=LINE('', #170, #180);
#182=EDGE_CURVE('', #171, #178, #181, .T.);
#183=ORIENTED_EDGE('', *, *, #182, .T.);
#184=CARTESIAN_POINT('', (0.0, -40.0, 5.0));
#185=VERTEX_POINT('', #184);
#186=DIRECTION('', (0.0, 0.0, 1.0));
#187=VECTOR('', #186, 170.0);
#188=LINE('', #184, #187);
#189=EDGE_CURVE('', #185, #178, #188, .T.);
#190=ORIENTED_EDGE('', *, *, #189, .F.);
#191=CARTESIAN_POINT('', (0.0, 0.0, 5.0));
#192=VERTEX_POINT('', #191);
#193=DIRECTION('', (0.0, -1.0, 0.0));
#194=VECTOR('', #193, 40.0);
#195=LINE('', #191, #194);
#196=EDGE_CURVE('', #192, #185, #195, .T.);
#197=ORIENTED_EDGE('', *, *, #196, .F.);
#198=CARTESIAN_POINT('', (0.0, 0.0, 81.0));
#199=VERTEX_POINT('', #198);
#200=DIRECTION('', (0.0, 0.0, 1.0));
#201=VECTOR('', #200, 76.0);
#202=LINE('', #191, #201);
#203=EDGE_CURVE('', #192, #199, #202, .T.);
#204=ORIENTED_EDGE('', *, *, #203, .T.);
#205=CARTESIAN_POINT('', (0.0, -10.0, 81.0));
#206=VERTEX_POINT('', #205);
#207=DIRECTION('', (0.0, -1.0, 0.0));
#208=VECTOR('', #207, 10.0);
#209=LINE('', #198, #208);
#210=EDGE_CURVE('', #199, #206, #209, .T.);
#211=ORIENTED_EDGE('', *, *, #210, .T.);
#212=CARTESIAN_POINT('', (0.0, -10.0, 74.0));
#213=VERTEX_POINT('', #212);
#214=DIRECTION('', (0.0, 0.0, 1.0));
#215=VECTOR('', #214, 7.0);
#216=LINE('', #212, #215);
#217=EDGE_CURVE('', #213, #206, #216, .T.);
#218=ORIENTED_EDGE('', *, *, #217, .F.);
#219=CARTESIAN_POINT('', (0.0, -30.0, 74.0));
#220=VERTEX_POINT('', #219);
#221=DIRECTION('', (0.0, 1.0, 0.0));
#222=VECTOR('', #221, 20.0);
#223=LINE('', #219, #222);
#224=EDGE_CURVE('', #220, #213, #223, .T.);
#225=ORIENTED_EDGE('', *, *, #224, .F.);
#226=DIRECTION('', (0.0, 0.0, 1.0));
#227=VECTOR('', #226, 32.0);
#228=LINE('', #219, #227);
#229=EDGE_CURVE('', #220, #148, #228, .T.);
#230=ORIENTED_EDGE('', *, *, #229, .T.);
#231=EDGE_LOOP('', (#155, #162, #169, #176, #183, #190, #197, #204, #211, #218, #225, #230));
#232=FACE_BOUND('', #231, .F.);

```

```

#233=CARTESIAN_POINT('', (0.0,0.0,0.0));
#234=DIRECTION('', (-1.0,0.0,0.0));
#235=DIRECTION('', (0.0,-1.0,0.0));
#236=AXIS2_PLACEMENT_3D('', #233, #234, #235);
#237=PLANE('', #236);
#238=ADVANCED_FACE('', (#232), #237, .T.);
#239=SHAPE_ASPECT('featured shape: face 3', $, #38, .T.);
#240=PROPERTY_DEFINITION('', $, #239);
#241=FACE_SHAPE_REPRESENTATION('', (#238), #29);
#242=PROPERTY_DEFINITION_REPRESENTATION(#240, #241);
#243=CARTESIAN_POINT('', (7.0,-50.0,50.000000000000007));
#244=VERTEX_POINT('', #243);
#245=CARTESIAN_POINT('', (45.0,-50.0,50.000000000000007));
#246=VERTEX_POINT('', #245);
#247=DIRECTION('', (1.0,0.0,0.0));
#248=VECTOR('', #247, 38.0);
#249=LINE('', #243, #248);
#250=EDGE_CURVE('', #244, #246, #249, .T.);
#251=ORIENTED_EDGE('', *, *, #250, .T.);
#252=CARTESIAN_POINT('', (50.0,-50.0,45.000000000000007));
#253=VERTEX_POINT('', #252);
#254=CARTESIAN_POINT('', (45.0,-50.0,45.000000000000007));
#255=DIRECTION('', (0.0,-1.0,0.0));
#256=DIRECTION('', (1.0,0.0,0.0));
#257=AXIS2_PLACEMENT_3D('', #254, #255, #256);
#258=CIRCLE('', #257, 5.0);
#259=EDGE_CURVE('', #253, #246, #258, .T.);
#260=ORIENTED_EDGE('', *, *, #259, .F.);
#261=CARTESIAN_POINT('', (50.0,-50.0,7.0));
#262=VERTEX_POINT('', #261);
#263=DIRECTION('', (0.0,0.0,1.0));
#264=VECTOR('', #263, 38.000000000000007);
#265=LINE('', #261, #264);
#266=EDGE_CURVE('', #262, #253, #265, .T.);
#267=ORIENTED_EDGE('', *, *, #266, .F.);
#268=CARTESIAN_POINT('', (45.0,-50.0,2.0));
#269=VERTEX_POINT('', #268);
#270=CARTESIAN_POINT('', (45.0,-50.0,7.0));
#271=DIRECTION('', (0.0,-1.0,0.0));
#272=DIRECTION('', (0.0,0.0,-1.0));
#273=AXIS2_PLACEMENT_3D('', #270, #271, #272);
#274=CIRCLE('', #273, 5.0);
#275=EDGE_CURVE('', #269, #262, #274, .T.);
#276=ORIENTED_EDGE('', *, *, #275, .F.);
#277=CARTESIAN_POINT('', (7.0,-50.0,2.0));
#278=VERTEX_POINT('', #277);
#279=DIRECTION('', (1.0,0.0,0.0));
#280=VECTOR('', #279, 38.0);
#281=LINE('', #277, #280);
#282=EDGE_CURVE('', #278, #269, #281, .T.);
#283=ORIENTED_EDGE('', *, *, #282, .F.);
#284=CARTESIAN_POINT('', (2.0,-50.0,7.0));
#285=VERTEX_POINT('', #284);

```

ISO 10303-223:2008 (E)

```

#286=CARTESIAN_POINT('', (7.0, -50.0, 7.0));
#287=DIRECTION('', (0.0, -1.0, 0.0));
#288=DIRECTION('', (-1.0, 0.0, 0.0));
#289=AXIS2_PLACEMENT_3D('', #286, #287, #288);
#290=CIRCLE('', #289, 5.0);
#291=EDGE_CURVE('', #285, #278, #290, .T.);
#292=ORIENTED_EDGE('', *, *, #291, .F.);
#293=CARTESIAN_POINT('', (2.0, -50.0, 45.000000000000007));
#294=VERTEX_POINT('', #293);
#295=DIRECTION('', (0.0, 0.0, 1.0));
#296=VECTOR('', #295, 38.000000000000007);
#297=LINE('', #284, #296);
#298=EDGE_CURVE('', #285, #294, #297, .T.);
#299=ORIENTED_EDGE('', *, *, #298, .T.);
#300=CARTESIAN_POINT('', (7.0, -50.0, 45.000000000000007));
#301=DIRECTION('', (0.0, 1.0, 0.0));
#302=DIRECTION('', (-1.0, 0.0, 0.0));
#303=AXIS2_PLACEMENT_3D('', #300, #301, #302);
#304=CIRCLE('', #303, 5.0);
#305=EDGE_CURVE('', #294, #244, #304, .T.);
#306=ORIENTED_EDGE('', *, *, #305, .T.);
#307=EDGE_LOOP('', (#251, #260, #267, #276, #283, #292, #299, #306));
#308=FACE_BOUND('', #307, .F.);
#309=CARTESIAN_POINT('', (34.0, -50.0, 27.500000000000000));
#310=VERTEX_POINT('', #309);
#311=CARTESIAN_POINT('', (16.0, -50.0, 27.500000000000000));
#312=VERTEX_POINT('', #311);
#313=CARTESIAN_POINT('', (25.0, -50.0, 27.500000000000000));
#314=DIRECTION('', (0.0, 1.0, 0.0));
#315=DIRECTION('', (1.0, 0.0, 0.0));
#316=AXIS2_PLACEMENT_3D('', #313, #314, #315);
#317=CIRCLE('', #316, 9.0);
#318=EDGE_CURVE('', #310, #312, #317, .T.);
#319=ORIENTED_EDGE('', *, *, #318, .F.);
#320=CARTESIAN_POINT('', (25.0, -50.0, 27.500000000000000));
#321=DIRECTION('', (0.0, 1.0, 0.0));
#322=DIRECTION('', (-1.0, 0.0, 0.0));
#323=AXIS2_PLACEMENT_3D('', #320, #321, #322);
#324=CIRCLE('', #323, 9.0);
#325=EDGE_CURVE('', #312, #310, #324, .T.);
#326=ORIENTED_EDGE('', *, *, #325, .F.);
#327=EDGE_LOOP('', (#319, #326));
#328=FACE_BOUND('', #327, .F.);
#329=CARTESIAN_POINT('', (0.0, -50.0, 0.0));
#330=DIRECTION('', (0.0, -1.0, 0.0));
#331=DIRECTION('', (1.0, 0.0, 0.0));
#332=AXIS2_PLACEMENT_3D('', #329, #330, #331);
#333=PLANE('', #332);
#334=ADVANCED_FACE('', (#308, #328), #333, .T.);
#335=SHAPE_ASPECT('featured shape: face 4', $, #38, .T.);
#336=PROPERTY_DEFINITION('', $, #335);
#337=FACE_SHAPE_REPRESENTATION('', (#334), #29);
#338=PROPERTY_DEFINITION_REPRESENTATION(#336, #337);

```

```

#339=CARTESIAN_POINT('', (180.000000000000028, -50.0, 7.0));
#340=VERTEX_POINT('', #339);
#341=CARTESIAN_POINT('', (180.000000000000028, -50.0, 45.0));
#342=VERTEX_POINT('', #341);
#343=DIRECTION('', (0.0, 0.0, 1.0));
#344=VECTOR('', #343, 38.0);
#345=LINE('', #339, #344);
#346=EDGE_CURVE('', #340, #342, #345, .T.);
#347=ORIENTED_EDGE('', *, *, #346, .T.);
#348=CARTESIAN_POINT('', (185.000000000000028, -50.0, 50.0));
#349=VERTEX_POINT('', #348);
#350=CARTESIAN_POINT('', (185.000000000000028, -50.0, 45.0));
#351=DIRECTION('', (0.0, -1.0, 0.0));
#352=DIRECTION('', (0.0, 0.0, 1.0));
#353=AXIS2_PLACEMENT_3D('', #350, #351, #352);
#354=CIRCLE('', #353, 5.0);
#355=EDGE_CURVE('', #349, #342, #354, .T.);
#356=ORIENTED_EDGE('', *, *, #355, .F.);
#357=CARTESIAN_POINT('', (223.0, -50.0, 50.0));
#358=VERTEX_POINT('', #357);
#359=DIRECTION('', (-1.0, 0.0, 0.0));
#360=VECTOR('', #359, 37.999999999999972);
#361=LINE('', #357, #360);
#362=EDGE_CURVE('', #358, #349, #361, .T.);
#363=ORIENTED_EDGE('', *, *, #362, .F.);
#364=CARTESIAN_POINT('', (228.0, -50.0, 45.0));
#365=VERTEX_POINT('', #364);
#366=CARTESIAN_POINT('', (223.0, -50.0, 45.0));
#367=DIRECTION('', (0.0, -1.0, 0.0));
#368=DIRECTION('', (1.0, 0.0, 0.0));
#369=AXIS2_PLACEMENT_3D('', #366, #367, #368);
#370=CIRCLE('', #369, 5.0);
#371=EDGE_CURVE('', #365, #358, #370, .T.);
#372=ORIENTED_EDGE('', *, *, #371, .F.);
#373=CARTESIAN_POINT('', (228.0, -50.0, 7.0));
#374=VERTEX_POINT('', #373);
#375=DIRECTION('', (0.0, 0.0, 1.0));
#376=VECTOR('', #375, 38.0);
#377=LINE('', #373, #376);
#378=EDGE_CURVE('', #374, #365, #377, .T.);
#379=ORIENTED_EDGE('', *, *, #378, .F.);
#380=CARTESIAN_POINT('', (223.0, -50.0, 2.0));
#381=VERTEX_POINT('', #380);
#382=CARTESIAN_POINT('', (223.0, -50.0, 7.0));
#383=DIRECTION('', (0.0, 1.0, 0.0));
#384=DIRECTION('', (1.0, 0.0, 0.0));
#385=AXIS2_PLACEMENT_3D('', #382, #383, #384);
#386=CIRCLE('', #385, 5.0);
#387=EDGE_CURVE('', #374, #381, #386, .T.);
#388=ORIENTED_EDGE('', *, *, #387, .T.);
#389=CARTESIAN_POINT('', (185.000000000000028, -50.0, 2.0));
#390=VERTEX_POINT('', #389);
#391=DIRECTION('', (1.0, 0.0, 0.0));

```

```

#392=VECTOR('', #391, 37.999999999999972);
#393=LINE('', #389, #392);
#394=EDGE_CURVE('', #390, #381, #393, .T.);
#395=ORIENTED_EDGE('', *, *, #394, .F.);
#396=CARTESIAN_POINT('', (185.000000000000028, -50.0, 7.0));
#397=DIRECTION('', (0.0, 1.0, 0.0));
#398=DIRECTION('', (0.0, 0.0, -1.0));
#399=AXIS2_PLACEMENT_3D('', #396, #397, #398);
#400=CIRCLE('', #399, 5.0);
#401=EDGE_CURVE('', #390, #340, #400, .T.);
#402=ORIENTED_EDGE('', *, *, #401, .T.);
#403=EDGE_LOOP('', (#347, #356, #363, #372, #379, #388, #395, #402));
#404=FACE_BOUND('', #403, .F.);
#405=CARTESIAN_POINT('', (214.0, -50.0, 27.500000000000000));
#406=VERTEX_POINT('', #405);
#407=CARTESIAN_POINT('', (196.0, -50.0, 27.500000000000000));
#408=VERTEX_POINT('', #407);
#409=CARTESIAN_POINT('', (205.0, -50.0, 27.500000000000000));
#410=DIRECTION('', (0.0, 1.0, 0.0));
#411=DIRECTION('', (1.0, 0.0, 0.0));
#412=AXIS2_PLACEMENT_3D('', #409, #410, #411);
#413=CIRCLE('', #412, 9.0);
#414=EDGE_CURVE('', #406, #408, #413, .T.);
#415=ORIENTED_EDGE('', *, *, #414, .F.);
#416=CARTESIAN_POINT('', (205.0, -50.0, 27.500000000000000));
#417=DIRECTION('', (0.0, 1.0, 0.0));
#418=DIRECTION('', (-1.0, 0.0, 0.0));
#419=AXIS2_PLACEMENT_3D('', #416, #417, #418);
#420=CIRCLE('', #419, 9.0);
#421=EDGE_CURVE('', #408, #406, #420, .T.);
#422=ORIENTED_EDGE('', *, *, #421, .F.);
#423=EDGE_LOOP('', (#415, #422));
#424=FACE_BOUND('', #423, .F.);
#425=ADVANCED_FACE('', (#404, #424), #333, .T.);
#426=SHAPE_ASPECT('featured shape: face 5', $, #38, .T.);
#427=PROPERTY_DEFINITION('', $, #426);
#428=FACE_SHAPE_REPRESENTATION('', (#425), #29);
#429=PROPERTY_DEFINITION_REPRESENTATION(#427, #428);
#430=CARTESIAN_POINT('', (223.0, -50.0, 130.000000000000028));
#431=VERTEX_POINT('', #430);
#432=CARTESIAN_POINT('', (184.999999999999972, -50.0, 130.000000000000028));
#433=VERTEX_POINT('', #432);
#434=DIRECTION('', (-1.0, 0.0, 0.0));
#435=VECTOR('', #434, 38.000000000000028);
#436=LINE('', #430, #435);
#437=EDGE_CURVE('', #431, #433, #436, .T.);
#438=ORIENTED_EDGE('', *, *, #437, .T.);
#439=CARTESIAN_POINT('', (179.999999999999972, -50.0, 135.000000000000028));
#440=VERTEX_POINT('', #439);
#441=CARTESIAN_POINT('', (184.999999999999972, -50.0, 135.000000000000028));
#442=DIRECTION('', (0.0, -1.0, 0.0));
#443=DIRECTION('', (-1.0, 0.0, 0.0));
#444=AXIS2_PLACEMENT_3D('', #441, #442, #443);

```



```

#445=CIRCLE('',#444,5.0);
#446=EDGE_CURVE('',#440,#433,#445,.T.);
#447=ORIENTED_EDGE('',*,*,#446,.F.);
#448=CARTESIAN_POINT('',(179.999999999999972,-50.0,173.0));
#449=VERTEX_POINT('',#448);
#450=DIRECTION('',(0.0,0.0,-1.0));
#451=VECTOR('',#450,37.999999999999972);
#452=LINE('',#448,#451);
#453=EDGE_CURVE('',#449,#440,#452,.T.);
#454=ORIENTED_EDGE('',*,*,#453,.F.);
#455=CARTESIAN_POINT('',(184.999999999999972,-50.0,178.0));
#456=VERTEX_POINT('',#455);
#457=CARTESIAN_POINT('',(184.999999999999972,-50.0,173.0));
#458=DIRECTION('',(0.0,-1.0,0.0));
#459=DIRECTION('',(0.0,0.0,1.0));
#460=AXIS2_PLACEMENT_3D('',#457,#458,#459);
#461=CIRCLE('',#460,5.0);
#462=EDGE_CURVE('',#456,#449,#461,.T.);
#463=ORIENTED_EDGE('',*,*,#462,.F.);
#464=CARTESIAN_POINT('',(223.0,-50.0,178.0));
#465=VERTEX_POINT('',#464);
#466=DIRECTION('',(1.0,0.0,0.0));
#467=VECTOR('',#466,38.000000000000028);
#468=LINE('',#455,#467);
#469=EDGE_CURVE('',#456,#465,#468,.T.);
#470=ORIENTED_EDGE('',*,*,#469,.T.);
#471=CARTESIAN_POINT('',(228.0,-50.0,173.0));
#472=VERTEX_POINT('',#471);
#473=CARTESIAN_POINT('',(223.0,-50.0,173.0));
#474=DIRECTION('',(0.0,-1.0,0.0));
#475=DIRECTION('',(1.0,0.0,0.0));
#476=AXIS2_PLACEMENT_3D('',#473,#474,#475);
#477=CIRCLE('',#476,5.0);
#478=EDGE_CURVE('',#472,#465,#477,.T.);
#479=ORIENTED_EDGE('',*,*,#478,.F.);
#480=CARTESIAN_POINT('',(228.0,-50.0,135.000000000000028));
#481=VERTEX_POINT('',#480);
#482=DIRECTION('',(0.0,0.0,1.0));
#483=VECTOR('',#482,37.999999999999972);
#484=LINE('',#480,#483);
#485=EDGE_CURVE('',#481,#472,#484,.T.);
#486=ORIENTED_EDGE('',*,*,#485,.F.);
#487=CARTESIAN_POINT('',(223.0,-50.0,135.000000000000028));
#488=DIRECTION('',(0.0,1.0,0.0));
#489=DIRECTION('',(1.0,0.0,0.0));
#490=AXIS2_PLACEMENT_3D('',#487,#488,#489);
#491=CIRCLE('',#490,5.0);
#492=EDGE_CURVE('',#481,#431,#491,.T.);
#493=ORIENTED_EDGE('',*,*,#492,.T.);
#494=EDGE_LOOP('',(#438,#447,#454,#463,#470,#479,#486,#493));
#495=FACE_BOUND('',#494,.F.);
#496=CARTESIAN_POINT('',(214.0,-50.0,152.500000000000000));
#497=VERTEX_POINT('',#496);

```

```

#498=CARTESIAN_POINT('', (196.0, -50.0, 152.50000000000000));
#499=VERTEX_POINT('', #498);
#500=CARTESIAN_POINT('', (205.0, -50.0, 152.50000000000000));
#501=DIRECTION('', (0.0, 1.0, 0.0));
#502=DIRECTION('', (1.0, 0.0, 0.0));
#503=AXIS2_PLACEMENT_3D('', #500, #501, #502);
#504=CIRCLE('', #503, 9.0);
#505=EDGE_CURVE('', #497, #499, #504, .T.);
#506=ORIENTED_EDGE('', *, *, #505, .F.);
#507=CARTESIAN_POINT('', (205.0, -50.0, 152.50000000000000));
#508=DIRECTION('', (0.0, 1.0, 0.0));
#509=DIRECTION('', (-1.0, 0.0, 0.0));
#510=AXIS2_PLACEMENT_3D('', #507, #508, #509);
#511=CIRCLE('', #510, 9.0);
#512=EDGE_CURVE('', #499, #497, #511, .T.);
#513=ORIENTED_EDGE('', *, *, #512, .F.);
#514=EDGE_LOOP('', (#506, #513));
#515=FACE_BOUND('', #514, .F.);
#516=ADVANCED_FACE('', (#495, #515), #333, .T.);
#517=SHAPE_ASPECT('featured shape: face 6', $, #38, .T.);
#518=PROPERTY_DEFINITION('', $, #517);
#519=FACE_SHAPE_REPRESENTATION('', (#516), #29);
#520=PROPERTY_DEFINITION_REPRESENTATION(#518, #519);
#521=CARTESIAN_POINT('', (49.999999999999972, -50.0, 173.0));
#522=VERTEX_POINT('', #521);
#523=CARTESIAN_POINT('', (49.999999999999972, -50.0, 134.999999999999972));
#524=VERTEX_POINT('', #523);
#525=DIRECTION('', (0.0, 0.0, -1.0));
#526=VECTOR('', #525, 38.000000000000028);
#527=LINE('', #521, #526);
#528=EDGE_CURVE('', #522, #524, #527, .T.);
#529=ORIENTED_EDGE('', *, *, #528, .T.);
#530=CARTESIAN_POINT('', (44.999999999999972, -50.0, 129.999999999999972));
#531=VERTEX_POINT('', #530);
#532=CARTESIAN_POINT('', (44.999999999999972, -50.0, 134.999999999999972));
#533=DIRECTION('', (0.0, -1.0, 0.0));
#534=DIRECTION('', (0.0, 0.0, -1.0));
#535=AXIS2_PLACEMENT_3D('', #532, #533, #534);
#536=CIRCLE('', #535, 5.0);
#537=EDGE_CURVE('', #531, #524, #536, .T.);
#538=ORIENTED_EDGE('', *, *, #537, .F.);
#539=CARTESIAN_POINT('', (7.0, -50.0, 129.999999999999972));
#540=VERTEX_POINT('', #539);
#541=DIRECTION('', (1.0, 0.0, 0.0));
#542=VECTOR('', #541, 37.999999999999972);
#543=LINE('', #539, #542);
#544=EDGE_CURVE('', #540, #531, #543, .T.);
#545=ORIENTED_EDGE('', *, *, #544, .F.);
#546=CARTESIAN_POINT('', (2.0, -50.0, 134.999999999999972));
#547=VERTEX_POINT('', #546);
#548=CARTESIAN_POINT('', (7.0, -50.0, 134.999999999999972));
#549=DIRECTION('', (0.0, -1.0, 0.0));
#550=DIRECTION('', (-1.0, 0.0, 0.0));

```

```

#551=AXIS2_PLACEMENT_3D('', #548, #549, #550);
#552=CIRCLE('', #551, 5.0);
#553=EDGE_CURVE('', #547, #540, #552, .T.);
#554=ORIENTED_EDGE('', *, *, #553, .F.);
#555=CARTESIAN_POINT('', (2.0, -50.0, 173.0));
#556=VERTEX_POINT('', #555);
#557=DIRECTION('', (0.0, 0.0, 1.0));
#558=VECTOR('', #557, 38.000000000000028);
#559=LINE('', #546, #558);
#560=EDGE_CURVE('', #547, #556, #559, .T.);
#561=ORIENTED_EDGE('', *, *, #560, .T.);
#562=CARTESIAN_POINT('', (7.0, -50.0, 178.0));
#563=VERTEX_POINT('', #562);
#564=CARTESIAN_POINT('', (7.0, -50.0, 173.0));
#565=DIRECTION('', (0.0, 1.0, 0.0));
#566=DIRECTION('', (-1.0, 0.0, 0.0));
#567=AXIS2_PLACEMENT_3D('', #564, #565, #566);
#568=CIRCLE('', #567, 5.0);
#569=EDGE_CURVE('', #556, #563, #568, .T.);
#570=ORIENTED_EDGE('', *, *, #569, .T.);
#571=CARTESIAN_POINT('', (44.999999999999972, -50.0, 178.0));
#572=VERTEX_POINT('', #571);
#573=DIRECTION('', (1.0, 0.0, 0.0));
#574=VECTOR('', #573, 37.999999999999972);
#575=LINE('', #562, #574);
#576=EDGE_CURVE('', #563, #572, #575, .T.);
#577=ORIENTED_EDGE('', *, *, #576, .T.);
#578=CARTESIAN_POINT('', (44.999999999999972, -50.0, 173.0));
#579=DIRECTION('', (0.0, 1.0, 0.0));
#580=DIRECTION('', (0.0, 0.0, 1.0));
#581=AXIS2_PLACEMENT_3D('', #578, #579, #580);
#582=CIRCLE('', #581, 5.0);
#583=EDGE_CURVE('', #572, #522, #582, .T.);
#584=ORIENTED_EDGE('', *, *, #583, .T.);
#585=EDGE_LOOP('', (#529, #538, #545, #554, #561, #570, #577, #584));
#586=FACE_BOUND('', #585, .F.);
#587=CARTESIAN_POINT('', (34.0, -50.0, 152.50000000000000));
#588=VERTEX_POINT('', #587);
#589=CARTESIAN_POINT('', (16.0, -50.0, 152.50000000000000));
#590=VERTEX_POINT('', #589);
#591=CARTESIAN_POINT('', (25.0, -50.0, 152.50000000000000));
#592=DIRECTION('', (0.0, 1.0, 0.0));
#593=DIRECTION('', (1.0, 0.0, 0.0));
#594=AXIS2_PLACEMENT_3D('', #591, #592, #593);
#595=CIRCLE('', #594, 9.0);
#596=EDGE_CURVE('', #588, #590, #595, .T.);
#597=ORIENTED_EDGE('', *, *, #596, .F.);
#598=CARTESIAN_POINT('', (25.0, -50.0, 152.50000000000000));
#599=DIRECTION('', (0.0, 1.0, 0.0));
#600=DIRECTION('', (-1.0, 0.0, 0.0));
#601=AXIS2_PLACEMENT_3D('', #598, #599, #600);
#602=CIRCLE('', #601, 9.0);
#603=EDGE_CURVE('', #590, #588, #602, .T.);

```

```

#604=ORIENTED_EDGE('',*,*,#603,.F.);
#605=EDGE_LOOP('',(#597,#604));
#606=FACE_BOUND('',#605,.F.);
#607=ADVANCED_FACE('',(#586,#606),#333,.T.);
#608=SHAPE_ASPECT('featured shape: face 7',$,#38,.T.);
#609=PROPERTY_DEFINITION('',$,#608);
#610=FACE_SHAPE_REPRESENTATION('',(#607),#29);
#611=PROPERTY_DEFINITION_REPRESENTATION(#609,#610);
#612=CARTESIAN_POINT('',(260.0,-40.0,55.0));
#613=VERTEX_POINT('',#612);
#614=CARTESIAN_POINT('',(260.0,-20.0,55.0));
#615=VERTEX_POINT('',#614);
#616=DIRECTION('',(0.0,1.0,0.0));
#617=VECTOR('',#616,20.0);
#618=LINE('',#612,#617);
#619=EDGE_CURVE('',#613,#615,#618,.T.);
#620=ORIENTED_EDGE('',*,*,#619,.F.);
#621=CARTESIAN_POINT('',(260.0,-40.0,125.0));
#622=VERTEX_POINT('',#621);
#623=DIRECTION('',(0.0,0.0,1.0));
#624=VECTOR('',#623,70.0);
#625=LINE('',#612,#624);
#626=EDGE_CURVE('',#613,#622,#625,.T.);
#627=ORIENTED_EDGE('',*,*,#626,.T.);
#628=CARTESIAN_POINT('',(260.0,-20.0,125.0));
#629=VERTEX_POINT('',#628);
#630=DIRECTION('',(0.0,1.0,0.0));
#631=VECTOR('',#630,20.0);
#632=LINE('',#621,#631);
#633=EDGE_CURVE('',#622,#629,#632,.T.);
#634=ORIENTED_EDGE('',*,*,#633,.T.);
#635=DIRECTION('',(0.0,0.0,1.0));
#636=VECTOR('',#635,70.0);
#637=LINE('',#614,#636);
#638=EDGE_CURVE('',#615,#629,#637,.T.);
#639=ORIENTED_EDGE('',*,*,#638,.F.);
#640=EDGE_LOOP('',(#620,#627,#634,#639));
#641=FACE_BOUND('',#640,.F.);
#642=CARTESIAN_POINT('',(260.0,-50.0,0.0));
#643=DIRECTION('',(1.0,0.0,0.0));
#644=DIRECTION('',(0.0,1.0,0.0));
#645=AXIS2_PLACEMENT_3D('',#642,#643,#644);
#646=PLANE('',#645);
#647=ADVANCED_FACE('',(#641),#646,.T.);
#648=SHAPE_ASPECT('featured shape: face 8',$,#38,.T.);
#649=PROPERTY_DEFINITION('',$,#648);
#650=FACE_SHAPE_REPRESENTATION('',(#647),#29);
#651=PROPERTY_DEFINITION_REPRESENTATION(#649,#650);
#652=CARTESIAN_POINT('',(117.999999999999943,0.0,135.0));
#653=VERTEX_POINT('',#652);
#654=DIRECTION('',(0.0,0.0,-1.0));
#655=VECTOR('',#654,45.0);
#656=LINE('',#88,#655);

```

```

#657=EDGE_CURVE('',#89,#653,#656,.T.);
#658=ORIENTED_EDGE('',*,*,#657,.T.);
#659=CARTESIAN_POINT('',(112.999999999999943,0.0,130.0));
#660=VERTEX_POINT('',#659);
#661=CARTESIAN_POINT('',(112.999999999999943,0.0,135.0));
#662=DIRECTION('',(0.0,1.0,0.0));
#663=DIRECTION('',(1.0,0.0,0.0));
#664=AXIS2_PLACEMENT_3D('',#661,#662,#663);
#665=CIRCLE('',#664,5.0);
#666=EDGE_CURVE('',#653,#660,#665,.T.);
#667=ORIENTED_EDGE('',*,*,#666,.T.);
#668=CARTESIAN_POINT('',(52.999999999999972,0.0,130.0));
#669=VERTEX_POINT('',#668);
#670=DIRECTION('',(-1.0,0.0,0.0));
#671=VECTOR('',#670,59.999999999999972);
#672=LINE('',#659,#671);
#673=EDGE_CURVE('',#660,#669,#672,.T.);
#674=ORIENTED_EDGE('',*,*,#673,.T.);
#675=CARTESIAN_POINT('',(47.999999999999972,0.0,135.0));
#676=VERTEX_POINT('',#675);
#677=CARTESIAN_POINT('',(52.999999999999979,0.0,135.0));
#678=DIRECTION('',(0.0,1.0,0.0));
#679=DIRECTION('',(-1.421085E-15,0.0,-1.0));
#680=AXIS2_PLACEMENT_3D('',#677,#678,#679);
#681=CIRCLE('',#680,5.0);
#682=EDGE_CURVE('',#669,#676,#681,.T.);
#683=ORIENTED_EDGE('',*,*,#682,.T.);
#684=DIRECTION('',(0.0,0.0,-1.0));
#685=VECTOR('',#684,45.0);
#686=LINE('',#123,#685);
#687=EDGE_CURVE('',#124,#676,#686,.T.);
#688=ORIENTED_EDGE('',*,*,#687,.F.);
#689=ORIENTED_EDGE('',*,*,#128,.T.);
#690=CARTESIAN_POINT('',(5.0,0.0,175.0));
#691=DIRECTION('',(0.0,-1.0,0.0));
#692=DIRECTION('',(0.0,0.0,1.0));
#693=AXIS2_PLACEMENT_3D('',#690,#691,#692);
#694=CIRCLE('',#693,5.0);
#695=EDGE_CURVE('',#117,#171,#694,.T.);
#696=ORIENTED_EDGE('',*,*,#695,.T.);
#697=ORIENTED_EDGE('',*,*,#175,.F.);
#698=CARTESIAN_POINT('',(49.999999694730889,0.0,99.0));
#699=VERTEX_POINT('',#698);
#700=DIRECTION('',(1.0,0.0,0.0));
#701=VECTOR('',#700,49.999999694730889);
#702=LINE('',#163,#701);
#703=EDGE_CURVE('',#164,#699,#702,.T.);
#704=ORIENTED_EDGE('',*,*,#703,.T.);
#705=CARTESIAN_POINT('',(50.0,0.0,81.0));
#706=VERTEX_POINT('',#705);
#707=CARTESIAN_POINT('',(50.0,0.0,90.0));
#708=DIRECTION('',(0.0,-1.0,0.0));
#709=DIRECTION('',(0.0,0.0,-1.0));

```

ISO 10303-223:2008 (E)

```

#710=AXIS2_PLACEMENT_3D('', #707, #708, #709);
#711=CIRCLE('', #710, 9.0);
#712=EDGE_CURVE('', #706, #699, #711, .T.);
#713=ORIENTED_EDGE('', *, *, #712, .F.);
#714=DIRECTION('', (-1.0, 0.0, 0.0));
#715=VECTOR('', #714, 50.0);
#716=LINE('', #705, #715);
#717=EDGE_CURVE('', #706, #199, #716, .T.);
#718=ORIENTED_EDGE('', *, *, #717, .T.);
#719=ORIENTED_EDGE('', *, *, #203, .F.);
#720=CARTESIAN_POINT('', (5.0, 0.0, 5.0));
#721=DIRECTION('', (0.0, 1.0, 0.0));
#722=DIRECTION('', (0.0, 0.0, -1.0));
#723=AXIS2_PLACEMENT_3D('', #720, #721, #722);
#724=CIRCLE('', #723, 5.0);
#725=EDGE_CURVE('', #56, #192, #724, .T.);
#726=ORIENTED_EDGE('', *, *, #725, .F.);
#727=ORIENTED_EDGE('', *, *, #60, .F.);
#728=CARTESIAN_POINT('', (230.0, 0.0, 5.0));
#729=VERTEX_POINT('', #728);
#730=CARTESIAN_POINT('', (225.0, 0.0, 5.0));
#731=DIRECTION('', (0.0, 1.0, 0.0));
#732=DIRECTION('', (1.0, 0.0, 0.0));
#733=AXIS2_PLACEMENT_3D('', #730, #731, #732);
#734=CIRCLE('', #733, 5.0);
#735=EDGE_CURVE('', #729, #49, #734, .T.);
#736=ORIENTED_EDGE('', *, *, #735, .F.);
#737=CARTESIAN_POINT('', (230.0, 0.0, 85.0));
#738=VERTEX_POINT('', #737);
#739=DIRECTION('', (0.0, 0.0, 1.0));
#740=VECTOR('', #739, 80.0);
#741=LINE('', #728, #740);
#742=EDGE_CURVE('', #729, #738, #741, .T.);
#743=ORIENTED_EDGE('', *, *, #742, .T.);
#744=CARTESIAN_POINT('', (210.0, 0.0, 85.0));
#745=VERTEX_POINT('', #744);
#746=DIRECTION('', (-1.0, 0.0, 0.0));
#747=VECTOR('', #746, 20.0);
#748=LINE('', #737, #747);
#749=EDGE_CURVE('', #738, #745, #748, .T.);
#750=ORIENTED_EDGE('', *, *, #749, .T.);
#751=CARTESIAN_POINT('', (210.0, 0.0, 95.0));
#752=VERTEX_POINT('', #751);
#753=CARTESIAN_POINT('', (210.0, 0.0, 90.0));
#754=DIRECTION('', (0.0, -1.0, 0.0));
#755=DIRECTION('', (0.0, 0.0, 1.0));
#756=AXIS2_PLACEMENT_3D('', #753, #754, #755);
#757=CIRCLE('', #756, 5.0);
#758=EDGE_CURVE('', #752, #745, #757, .T.);
#759=ORIENTED_EDGE('', *, *, #758, .F.);
#760=CARTESIAN_POINT('', (230.0, 0.0, 95.0));
#761=VERTEX_POINT('', #760);
#762=DIRECTION('', (-1.0, 0.0, 0.0));

```

```

#763=VECTOR('',#762,20.0);
#764=LINE('',#760,#763);
#765=EDGE_CURVE('',#761,#752,#764,.T.);
#766=ORIENTED_EDGE('',*,*,#765,.F.);
#767=CARTESIAN_POINT('',(230.0,0.0,175.0));
#768=VERTEX_POINT('',#767);
#769=DIRECTION('',(0.0,0.0,1.0));
#770=VECTOR('',#769,80.0);
#771=LINE('',#760,#770);
#772=EDGE_CURVE('',#761,#768,#771,.T.);
#773=ORIENTED_EDGE('',*,*,#772,.T.);
#774=CARTESIAN_POINT('',(225.0,0.0,175.0));
#775=DIRECTION('',(0.0,-1.0,0.0));
#776=DIRECTION('',(1.0,0.0,0.0));
#777=AXIS2_PLACEMENT_3D('',#774,#775,#776);
#778=CIRCLE('',#777,5.0);
#779=EDGE_CURVE('',#768,#96,#778,.T.);
#780=ORIENTED_EDGE('',*,*,#779,.T.);
#781=ORIENTED_EDGE('',*,*,#100,.T.);
#782=EDGE_LOOP('',(#658,#667,#674,#683,#688,#689,#696,#697,#704,#713,#718,#719,#726,#727,#736,#743,#750,#759,#766,#773,#780,#781));
#783=FACE_BOUND('',#782,.F.);
#784=CARTESIAN_POINT('',(180.0,0.0,90.0));
#785=VERTEX_POINT('',#784);
#786=CARTESIAN_POINT('',(140.0,0.0,90.0));
#787=VERTEX_POINT('',#786);
#788=CARTESIAN_POINT('',(160.0,0.0,90.0));
#789=DIRECTION('',(0.0,-1.0,0.0));
#790=DIRECTION('',(1.0,0.0,0.0));
#791=AXIS2_PLACEMENT_3D('',#788,#789,#790);
#792=CIRCLE('',#791,20.0);
#793=EDGE_CURVE('',#785,#787,#792,.T.);
#794=ORIENTED_EDGE('',*,*,#793,.F.);
#795=CARTESIAN_POINT('',(160.0,0.0,90.0));
#796=DIRECTION('',(0.0,-1.0,0.0));
#797=DIRECTION('',(-1.0,0.0,0.0));
#798=AXIS2_PLACEMENT_3D('',#795,#796,#797);
#799=CIRCLE('',#798,20.0);
#800=EDGE_CURVE('',#787,#785,#799,.T.);
#801=ORIENTED_EDGE('',*,*,#800,.F.);
#802=EDGE_LOOP('',(#794,#801));
#803=FACE_BOUND('',#802,.F.);
#804=CARTESIAN_POINT('',(172.50000000000000,0.0,30.0));
#805=VERTEX_POINT('',#804);
#806=CARTESIAN_POINT('',(147.50000000000000,0.0,30.0));
#807=VERTEX_POINT('',#806);
#808=CARTESIAN_POINT('',(160.0,0.0,30.0));
#809=DIRECTION('',(0.0,1.0,0.0));
#810=DIRECTION('',(1.0,0.0,0.0));
#811=AXIS2_PLACEMENT_3D('',#808,#809,#810);
#812=CIRCLE('',#811,12.500000000000000);
#813=EDGE_CURVE('',#805,#807,#812,.T.);
#814=ORIENTED_EDGE('',*,*,#813,.T.);

```

```

#815=CARTESIAN_POINT('', (160.0,0.0,30.0));
#816=DIRECTION('', (0.0,1.0,0.0));
#817=DIRECTION('', (-1.0,0.0,0.0));
#818=AXIS2_PLACEMENT_3D('', #815, #816, #817);
#819=CIRCLE('', #818, 12.500000000000000);
#820=EDGE_CURVE('', #807, #805, #819, .T.);
#821=ORIENTED_EDGE('', *, *, #820, .T.);
#822=EDGE_LOOP('', (#814, #821));
#823=FACE_BOUND('', #822, .F.);
#824=CARTESIAN_POINT('', (172.50000000000000, 0.0, 150.0));
#825=VERTEX_POINT('', #824);
#826=CARTESIAN_POINT('', (147.50000000000000, 0.0, 150.0));
#827=VERTEX_POINT('', #826);
#828=CARTESIAN_POINT('', (160.0, 0.0, 150.0));
#829=DIRECTION('', (0.0, 1.0, 0.0));
#830=DIRECTION('', (1.0, 0.0, 0.0));
#831=AXIS2_PLACEMENT_3D('', #828, #829, #830);
#832=CIRCLE('', #831, 12.500000000000000);
#833=EDGE_CURVE('', #825, #827, #832, .T.);
#834=ORIENTED_EDGE('', *, *, #833, .T.);
#835=CARTESIAN_POINT('', (160.0, 0.0, 150.0));
#836=DIRECTION('', (0.0, 1.0, 0.0));
#837=DIRECTION('', (-1.0, 0.0, 0.0));
#838=AXIS2_PLACEMENT_3D('', #835, #836, #837);
#839=CIRCLE('', #838, 12.500000000000000);
#840=EDGE_CURVE('', #827, #825, #839, .T.);
#841=ORIENTED_EDGE('', *, *, #840, .T.);
#842=EDGE_LOOP('', (#834, #841));
#843=FACE_BOUND('', #842, .F.);
#844=CARTESIAN_POINT('', (218.0, 0.0, 27.500000000000000));
#845=VERTEX_POINT('', #844);
#846=CARTESIAN_POINT('', (192.0, 0.0, 27.500000000000000));
#847=VERTEX_POINT('', #846);
#848=CARTESIAN_POINT('', (205.0, 0.0, 27.500000000000000));
#849=DIRECTION('', (0.0, 1.0, 0.0));
#850=DIRECTION('', (1.0, 0.0, 0.0));
#851=AXIS2_PLACEMENT_3D('', #848, #849, #850);
#852=CIRCLE('', #851, 13.0);
#853=EDGE_CURVE('', #845, #847, #852, .T.);
#854=ORIENTED_EDGE('', *, *, #853, .T.);
#855=CARTESIAN_POINT('', (205.0, 0.0, 27.500000000000000));
#856=DIRECTION('', (0.0, 1.0, 0.0));
#857=DIRECTION('', (-1.0, 0.0, 0.0));
#858=AXIS2_PLACEMENT_3D('', #855, #856, #857);
#859=CIRCLE('', #858, 13.0);
#860=EDGE_CURVE('', #847, #845, #859, .T.);
#861=ORIENTED_EDGE('', *, *, #860, .T.);
#862=EDGE_LOOP('', (#854, #861));
#863=FACE_BOUND('', #862, .F.);
#864=CARTESIAN_POINT('', (218.0, 0.0, 152.50000000000000));
#865=VERTEX_POINT('', #864);
#866=CARTESIAN_POINT('', (192.0, 0.0, 152.50000000000000));
#867=VERTEX_POINT('', #866);

```



```

#868=CARTESIAN_POINT('', (205.0,0.0,152.50000000000000));
#869=DIRECTION('', (0.0,1.0,0.0));
#870=DIRECTION('', (1.0,0.0,0.0));
#871=AXIS2_PLACEMENT_3D('', #868, #869, #870);
#872=CIRCLE('', #871, 13.0);
#873=EDGE_CURVE('', #865, #867, #872, .T.);
#874=ORIENTED_EDGE('', *, *, #873, .T.);
#875=CARTESIAN_POINT('', (205.0,0.0,152.50000000000000));
#876=DIRECTION('', (0.0,1.0,0.0));
#877=DIRECTION('', (-1.0,0.0,0.0));
#878=AXIS2_PLACEMENT_3D('', #875, #876, #877);
#879=CIRCLE('', #878, 13.0);
#880=EDGE_CURVE('', #867, #865, #879, .T.);
#881=ORIENTED_EDGE('', *, *, #880, .T.);
#882=EDGE_LOOP('', (#874, #881));
#883=FACE_BOUND('', #882, .F.);
#884=CARTESIAN_POINT('', (38.0,0.0,27.50000000000000));
#885=VERTEX_POINT('', #884);
#886=CARTESIAN_POINT('', (12.0,0.0,27.50000000000000));
#887=VERTEX_POINT('', #886);
#888=CARTESIAN_POINT('', (25.0,0.0,27.50000000000000));
#889=DIRECTION('', (0.0,1.0,0.0));
#890=DIRECTION('', (1.0,0.0,0.0));
#891=AXIS2_PLACEMENT_3D('', #888, #889, #890);
#892=CIRCLE('', #891, 13.0);
#893=EDGE_CURVE('', #885, #887, #892, .T.);
#894=ORIENTED_EDGE('', *, *, #893, .T.);
#895=CARTESIAN_POINT('', (25.0,0.0,27.50000000000000));
#896=DIRECTION('', (0.0,1.0,0.0));
#897=DIRECTION('', (-1.0,0.0,0.0));
#898=AXIS2_PLACEMENT_3D('', #895, #896, #897);
#899=CIRCLE('', #898, 13.0);
#900=EDGE_CURVE('', #887, #885, #899, .T.);
#901=ORIENTED_EDGE('', *, *, #900, .T.);
#902=EDGE_LOOP('', (#894, #901));
#903=FACE_BOUND('', #902, .F.);
#904=CARTESIAN_POINT('', (38.0,0.0,152.50000000000000));
#905=VERTEX_POINT('', #904);
#906=CARTESIAN_POINT('', (12.0,0.0,152.50000000000000));
#907=VERTEX_POINT('', #906);
#908=CARTESIAN_POINT('', (25.0,0.0,152.50000000000000));
#909=DIRECTION('', (0.0,1.0,0.0));
#910=DIRECTION('', (1.0,0.0,0.0));
#911=AXIS2_PLACEMENT_3D('', #908, #909, #910);
#912=CIRCLE('', #911, 13.0);
#913=EDGE_CURVE('', #905, #907, #912, .T.);
#914=ORIENTED_EDGE('', *, *, #913, .T.);
#915=CARTESIAN_POINT('', (25.0,0.0,152.50000000000000));
#916=DIRECTION('', (0.0,1.0,0.0));
#917=DIRECTION('', (-1.0,0.0,0.0));
#918=AXIS2_PLACEMENT_3D('', #915, #916, #917);
#919=CIRCLE('', #918, 13.0);
#920=EDGE_CURVE('', #907, #905, #919, .T.);

```

ISO 10303-223:2008 (E)

```

#921=ORIENTED_EDGE('',*,*,#920,.T.);
#922=EDGE_LOOP('',(#914,#921));
#923=FACE_BOUND('',#922,.F.);
#924=CARTESIAN_POINT('',(110.0,0.0,85.0));
#925=VERTEX_POINT('',#924);
#926=CARTESIAN_POINT('',(85.0,0.0,85.0));
#927=VERTEX_POINT('',#926);
#928=DIRECTION('',(-1.0,0.0,0.0));
#929=VECTOR('',#928,25.0);
#930=LINE('',#924,#929);
#931=EDGE_CURVE('',#925,#927,#930,.T.);
#932=ORIENTED_EDGE('',*,*,#931,.T.);
#933=CARTESIAN_POINT('',(85.0,0.0,95.0));
#934=VERTEX_POINT('',#933);
#935=CARTESIAN_POINT('',(85.0,0.0,90.0));
#936=DIRECTION('',(0.0,1.0,0.0));
#937=DIRECTION('',(0.0,0.0,-1.0));
#938=AXIS2_PLACEMENT_3D('',#935,#936,#937);
#939=CIRCLE('',#938,5.0);
#940=EDGE_CURVE('',#927,#934,#939,.T.);
#941=ORIENTED_EDGE('',*,*,#940,.T.);
#942=CARTESIAN_POINT('',(110.0,0.0,95.0));
#943=VERTEX_POINT('',#942);
#944=DIRECTION('',(1.0,0.0,0.0));
#945=VECTOR('',#944,25.0);
#946=LINE('',#933,#945);
#947=EDGE_CURVE('',#934,#943,#946,.T.);
#948=ORIENTED_EDGE('',*,*,#947,.T.);
#949=CARTESIAN_POINT('',(110.0,0.0,90.0));
#950=DIRECTION('',(0.0,1.0,0.0));
#951=DIRECTION('',(0.0,0.0,1.0));
#952=AXIS2_PLACEMENT_3D('',#949,#950,#951);
#953=CIRCLE('',#952,5.0);
#954=EDGE_CURVE('',#943,#925,#953,.T.);
#955=ORIENTED_EDGE('',*,*,#954,.T.);
#956=EDGE_LOOP('',(#932,#941,#948,#955));
#957=FACE_BOUND('',#956,.F.);
#958=CARTESIAN_POINT('',(192.568849999999998,0.0,120.0));
#959=VERTEX_POINT('',#958);
#960=CARTESIAN_POINT('',(187.4311500000000002,0.0,120.0));
#961=VERTEX_POINT('',#960);
#962=CARTESIAN_POINT('',(190.0,0.0,120.0));
#963=DIRECTION('',(0.0,-1.0,0.0));
#964=DIRECTION('',(1.0,0.0,0.0));
#965=AXIS2_PLACEMENT_3D('',#962,#963,#964);
#966=CIRCLE('',#965,2.5688500000000000);
#967=EDGE_CURVE('',#959,#961,#966,.T.);
#968=ORIENTED_EDGE('',*,*,#967,.F.);
#969=CARTESIAN_POINT('',(190.0,0.0,120.0));
#970=DIRECTION('',(0.0,-1.0,0.0));
#971=DIRECTION('',(-1.0,0.0,0.0));
#972=AXIS2_PLACEMENT_3D('',#969,#970,#971);
#973=CIRCLE('',#972,2.5688500000000000);

```

```

#974=EDGE_CURVE('',#961,#959,#973,.T.);
#975=ORIENTED_EDGE('',*,*,#974,.F.);
#976=EDGE_LOOP('',(#968,#975));
#977=FACE_BOUND('',#976,.F.);
#978=CARTESIAN_POINT('',(192.568849999999998,0.0,60.0));
#979=VERTEX_POINT('',#978);
#980=CARTESIAN_POINT('',(187.431150000000002,0.0,60.0));
#981=VERTEX_POINT('',#980);
#982=CARTESIAN_POINT('',(190.0,0.0,60.0));
#983=DIRECTION('',(0.0,-1.0,0.0));
#984=DIRECTION('',(1.0,0.0,0.0));
#985=AXIS2_PLACEMENT_3D('',#982,#983,#984);
#986=CIRCLE('',#985,2.568850000000000);
#987=EDGE_CURVE('',#979,#981,#986,.T.);
#988=ORIENTED_EDGE('',*,*,#987,.F.);
#989=CARTESIAN_POINT('',(190.0,0.0,60.0));
#990=DIRECTION('',(0.0,-1.0,0.0));
#991=DIRECTION('',(-1.0,0.0,0.0));
#992=AXIS2_PLACEMENT_3D('',#989,#990,#991);
#993=CIRCLE('',#992,2.568850000000000);
#994=EDGE_CURVE('',#981,#979,#993,.T.);
#995=ORIENTED_EDGE('',*,*,#994,.F.);
#996=EDGE_LOOP('',(#988,#995));
#997=FACE_BOUND('',#996,.F.);
#998=CARTESIAN_POINT('',(132.568849999999998,0.0,120.0));
#999=VERTEX_POINT('',#998);
#1000=CARTESIAN_POINT('',(127.431150000000002,0.0,120.0));
#1001=VERTEX_POINT('',#1000);
#1002=CARTESIAN_POINT('',(130.0,0.0,120.0));
#1003=DIRECTION('',(0.0,-1.0,0.0));
#1004=DIRECTION('',(1.0,0.0,0.0));
#1005=AXIS2_PLACEMENT_3D('',#1002,#1003,#1004);
#1006=CIRCLE('',#1005,2.568850000000000);
#1007=EDGE_CURVE('',#999,#1001,#1006,.T.);
#1008=ORIENTED_EDGE('',*,*,#1007,.F.);
#1009=CARTESIAN_POINT('',(130.0,0.0,120.0));
#1010=DIRECTION('',(0.0,-1.0,0.0));
#1011=DIRECTION('',(-1.0,0.0,0.0));
#1012=AXIS2_PLACEMENT_3D('',#1009,#1010,#1011);
#1013=CIRCLE('',#1012,2.568850000000000);
#1014=EDGE_CURVE('',#1001,#999,#1013,.T.);
#1015=ORIENTED_EDGE('',*,*,#1014,.F.);
#1016=EDGE_LOOP('',(#1008,#1015));
#1017=FACE_BOUND('',#1016,.F.);
#1018=CARTESIAN_POINT('',(132.568849999999998,0.0,60.0));
#1019=VERTEX_POINT('',#1018);
#1020=CARTESIAN_POINT('',(127.431150000000002,0.0,60.0));
#1021=VERTEX_POINT('',#1020);
#1022=CARTESIAN_POINT('',(130.0,0.0,60.0));
#1023=DIRECTION('',(0.0,-1.0,0.0));
#1024=DIRECTION('',(1.0,0.0,0.0));
#1025=AXIS2_PLACEMENT_3D('',#1022,#1023,#1024);
#1026=CIRCLE('',#1025,2.568850000000000);

```

```

#1027=EDGE_CURVE('', #1019, #1021, #1026, .T.);
#1028=ORIENTED_EDGE('', *, *, #1027, .F.);
#1029=CARTESIAN_POINT('', (130.0, 0.0, 60.0));
#1030=DIRECTION('', (0.0, -1.0, 0.0));
#1031=DIRECTION('', (-1.0, 0.0, 0.0));
#1032=AXIS2_PLACEMENT_3D('', #1029, #1030, #1031);
#1033=CIRCLE('', #1032, 2.5688500000000000);
#1034=EDGE_CURVE('', #1021, #1019, #1033, .T.);
#1035=ORIENTED_EDGE('', *, *, #1034, .F.);
#1036=EDGE_LOOP('', (#1028, #1035));
#1037=FACE_BOUND('', #1036, .F.);
#1038=CARTESIAN_POINT('', (100.0, 0.0, 10.0));
#1039=VERTEX_POINT('', #1038);
#1040=CARTESIAN_POINT('', (60.0, 0.0, 10.0));
#1041=VERTEX_POINT('', #1040);
#1042=DIRECTION('', (-1.0, 0.0, 0.0));
#1043=VECTOR('', #1042, 40.0);
#1044=LINE('', #1038, #1043);
#1045=EDGE_CURVE('', #1039, #1041, #1044, .T.);
#1046=ORIENTED_EDGE('', *, *, #1045, .T.);
#1047=CARTESIAN_POINT('', (50.0, 0.0, 20.0));
#1048=VERTEX_POINT('', #1047);
#1049=CARTESIAN_POINT('', (59.999999999999993, 1.421085E-14, 20.0));
#1050=DIRECTION('', (5.532395E-16, -1.0, 0.0));
#1051=DIRECTION('', (-1.0, -5.532395E-16, 0.0));
#1052=AXIS2_PLACEMENT_3D('', #1049, #1050, #1051);
#1053=CIRCLE('', #1052, 10.0);
#1054=EDGE_CURVE('', #1048, #1041, #1053, .T.);
#1055=ORIENTED_EDGE('', *, *, #1054, .F.);
#1056=CARTESIAN_POINT('', (50.0, 0.0, 50.0));
#1057=VERTEX_POINT('', #1056);
#1058=DIRECTION('', (0.0, 0.0, -1.0));
#1059=VECTOR('', #1058, 30.0);
#1060=LINE('', #1056, #1059);
#1061=EDGE_CURVE('', #1057, #1048, #1060, .T.);
#1062=ORIENTED_EDGE('', *, *, #1061, .F.);
#1063=CARTESIAN_POINT('', (60.0, 0.0, 60.0));
#1064=VERTEX_POINT('', #1063);
#1065=CARTESIAN_POINT('', (59.999999999999993, 3.552714E-15, 50.0));
#1066=DIRECTION('', (5.533360E-16, -1.0, 1.261554E-33));
#1067=DIRECTION('', (7.105427E-16, 3.944305E-31, 1.0));
#1068=AXIS2_PLACEMENT_3D('', #1065, #1066, #1067);
#1069=CIRCLE('', #1068, 10.0);
#1070=EDGE_CURVE('', #1064, #1057, #1069, .T.);
#1071=ORIENTED_EDGE('', *, *, #1070, .F.);
#1072=CARTESIAN_POINT('', (100.0, 0.0, 60.0));
#1073=VERTEX_POINT('', #1072);
#1074=DIRECTION('', (-1.0, 0.0, 0.0));
#1075=VECTOR('', #1074, 40.0);
#1076=LINE('', #1072, #1075);
#1077=EDGE_CURVE('', #1073, #1064, #1076, .T.);
#1078=ORIENTED_EDGE('', *, *, #1077, .F.);
#1079=CARTESIAN_POINT('', (110.0, 0.0, 50.0));

```

```

#1080=VERTEX_POINT('',#1079);
#1081=CARTESIAN_POINT('',(99.999999999999986,-1.387779E-17,50.0));
#1082=DIRECTION('',(0.0,1.0,0.0));
#1083=DIRECTION('',(1.421085E-15,0.0,1.0));
#1084=AXIS2_PLACEMENT_3D('',#1081,#1082,#1083);
#1085=CIRCLE('',#1084,10.0);
#1086=EDGE_CURVE('',#1073,#1080,#1085,.T.);
#1087=ORIENTED_EDGE('',*,*,#1086,.T.);
#1088=CARTESIAN_POINT('',(110.0,0.0,20.0));
#1089=VERTEX_POINT('',#1088);
#1090=DIRECTION('',(0.0,0.0,-1.0));
#1091=VECTOR('',#1090,30.0);
#1092=LINE('',#1079,#1091);
#1093=EDGE_CURVE('',#1080,#1089,#1092,.T.);
#1094=ORIENTED_EDGE('',*,*,#1093,.T.);
#1095=CARTESIAN_POINT('',(99.999999999999986,4.163336E-17,20.0));
#1096=DIRECTION('',(0.0,1.0,0.0));
#1097=DIRECTION('',(1.0,0.0,0.0));
#1098=AXIS2_PLACEMENT_3D('',#1095,#1096,#1097);
#1099=CIRCLE('',#1098,10.0);
#1100=EDGE_CURVE('',#1089,#1039,#1099,.T.);
#1101=ORIENTED_EDGE('',*,*,#1100,.T.);
#1102=EDGE_LOOP('',(#1046,#1055,#1062,#1071,#1078,#1087,#1094,#1101));
#1103=FACE_BOUND('',#1102,.F.);
#1104=CARTESIAN_POINT('',(260.0,0.0,0.0));
#1105=DIRECTION('',(0.0,1.0,0.0));
#1106=DIRECTION('',(-1.0,0.0,0.0));
#1107=AXIS2_PLACEMENT_3D('',#1104,#1105,#1106);
#1108=PLANE('',#1107);
#1109=ADVANCED_FACE('',(#783,#803,#823,#843,#863,#883,#903,#923,#957,#977,#997,#1017,#1037,#1103),#1108,.T.);
#1110=SHAPE_ASPECT('featured shape: face 9',$,#38,.T.);
#1111=PROPERTY_DEFINITION('',$,#1110);
#1112=FACE_SHAPE_REPRESENTATION('',(#1109),#29);
#1113=PROPERTY_DEFINITION_REPRESENTATION(#1111,#1112);
#1114=CARTESIAN_POINT('',(69.999999999999972,0.0,145.0));
#1115=VERTEX_POINT('',#1114);
#1116=CARTESIAN_POINT('',(95.999999999999972,0.0,145.0));
#1117=VERTEX_POINT('',#1116);
#1118=DIRECTION('',(1.0,0.0,0.0));
#1119=VECTOR('',#1118,26.0);
#1120=LINE('',#1114,#1119);
#1121=EDGE_CURVE('',#1115,#1117,#1120,.T.);
#1122=ORIENTED_EDGE('',*,*,#1121,.T.);
#1123=CARTESIAN_POINT('',(97.999999999999972,0.0,147.0));
#1124=VERTEX_POINT('',#1123);
#1125=CARTESIAN_POINT('',(95.999999999999972,0.0,147.0));
#1126=DIRECTION('',(0.0,1.0,0.0));
#1127=DIRECTION('',(1.0,0.0,0.0));
#1128=AXIS2_PLACEMENT_3D('',#1125,#1126,#1127);
#1129=CIRCLE('',#1128,2.0);
#1130=EDGE_CURVE('',#1124,#1117,#1129,.T.);
#1131=ORIENTED_EDGE('',*,*,#1130,.F.);

```

```

#1132=CARTESIAN_POINT('', (97.999999999999972, 0.0, 163.0));
#1133=VERTEX_POINT('', #1132);
#1134=DIRECTION('', (0.0, 0.0, 1.0));
#1135=VECTOR('', #1134, 16.0);
#1136=LINE('', #1123, #1135);
#1137=EDGE_CURVE('', #1124, #1133, #1136, .T.);
#1138=ORIENTED_EDGE('', *, *, #1137, .T.);
#1139=CARTESIAN_POINT('', (95.999999999999972, 0.0, 165.0));
#1140=VERTEX_POINT('', #1139);
#1141=CARTESIAN_POINT('', (95.999999999999972, 0.0, 163.0));
#1142=DIRECTION('', (0.0, 1.0, 0.0));
#1143=DIRECTION('', (0.0, 0.0, 1.0));
#1144=AXIS2_PLACEMENT_3D('', #1141, #1142, #1143);
#1145=CIRCLE('', #1144, 2.0);
#1146=EDGE_CURVE('', #1140, #1133, #1145, .T.);
#1147=ORIENTED_EDGE('', *, *, #1146, .F.);
#1148=CARTESIAN_POINT('', (69.999999999999972, 0.0, 165.0));
#1149=VERTEX_POINT('', #1148);
#1150=DIRECTION('', (-1.0, 0.0, 0.0));
#1151=VECTOR('', #1150, 26.0);
#1152=LINE('', #1139, #1151);
#1153=EDGE_CURVE('', #1140, #1149, #1152, .T.);
#1154=ORIENTED_EDGE('', *, *, #1153, .T.);
#1155=CARTESIAN_POINT('', (67.999999999999972, 0.0, 163.0));
#1156=VERTEX_POINT('', #1155);
#1157=CARTESIAN_POINT('', (69.999999999999972, 0.0, 163.0));
#1158=DIRECTION('', (0.0, 1.0, 0.0));
#1159=DIRECTION('', (-1.0, 0.0, 0.0));
#1160=AXIS2_PLACEMENT_3D('', #1157, #1158, #1159);
#1161=CIRCLE('', #1160, 2.0);
#1162=EDGE_CURVE('', #1156, #1149, #1161, .T.);
#1163=ORIENTED_EDGE('', *, *, #1162, .F.);
#1164=CARTESIAN_POINT('', (67.999999999999972, 0.0, 147.0));
#1165=VERTEX_POINT('', #1164);
#1166=DIRECTION('', (0.0, 0.0, -1.0));
#1167=VECTOR('', #1166, 16.0);
#1168=LINE('', #1155, #1167);
#1169=EDGE_CURVE('', #1156, #1165, #1168, .T.);
#1170=ORIENTED_EDGE('', *, *, #1169, .T.);
#1171=CARTESIAN_POINT('', (69.999999999999972, 0.0, 147.0));
#1172=DIRECTION('', (0.0, 1.0, 0.0));
#1173=DIRECTION('', (0.0, 0.0, -1.0));
#1174=AXIS2_PLACEMENT_3D('', #1171, #1172, #1173);
#1175=CIRCLE('', #1174, 2.0);
#1176=EDGE_CURVE('', #1115, #1165, #1175, .T.);
#1177=ORIENTED_EDGE('', *, *, #1176, .F.);
#1178=EDGE_LOOP('', (#1122, #1131, #1138, #1147, #1154, #1163, #1170, #1177));
#1179=FACE_BOUND('', #1178, .F.);
#1180=ADVANCED_FACE('', (#1179), #1108, .T.);
#1181=SHAPE_ASPECT('featured shape: face 10', $, #38, .T.);
#1182=PROPERTY_DEFINITION('', $, #1181);
#1183=FACE_SHAPE_REPRESENTATION('', (#1180), #29);
#1184=PROPERTY_DEFINITION_REPRESENTATION(#1182, #1183);

```

```

#1185=CARTESIAN_POINT('', (230.0, -10.0, 95.0));
#1186=VERTEX_POINT('', #1185);
#1187=DIRECTION('', (0.0, -1.0, 0.0));
#1188=VECTOR('', #1187, 10.0);
#1189=LINE('', #760, #1188);
#1190=EDGE_CURVE('', #761, #1186, #1189, .T.);
#1191=ORIENTED_EDGE('', *, *, #1190, .T.);
#1192=CARTESIAN_POINT('', (230.0, -10.0, 85.0));
#1193=VERTEX_POINT('', #1192);
#1194=DIRECTION('', (0.0, 0.0, -1.0));
#1195=VECTOR('', #1194, 10.0);
#1196=LINE('', #1185, #1195);
#1197=EDGE_CURVE('', #1186, #1193, #1196, .T.);
#1198=ORIENTED_EDGE('', *, *, #1197, .T.);
#1199=DIRECTION('', (0.0, -1.0, 0.0));
#1200=VECTOR('', #1199, 10.0);
#1201=LINE('', #737, #1200);
#1202=EDGE_CURVE('', #738, #1193, #1201, .T.);
#1203=ORIENTED_EDGE('', *, *, #1202, .F.);
#1204=ORIENTED_EDGE('', *, *, #742, .F.);
#1205=CARTESIAN_POINT('', (230.0, -40.0, 5.0));
#1206=VERTEX_POINT('', #1205);
#1207=DIRECTION('', (0.0, -1.0, 0.0));
#1208=VECTOR('', #1207, 40.0);
#1209=LINE('', #728, #1208);
#1210=EDGE_CURVE('', #729, #1206, #1209, .T.);
#1211=ORIENTED_EDGE('', *, *, #1210, .T.);
#1212=CARTESIAN_POINT('', (230.0, -40.0, 40.0));
#1213=VERTEX_POINT('', #1212);
#1214=DIRECTION('', (0.0, 0.0, 1.0));
#1215=VECTOR('', #1214, 35.0);
#1216=LINE('', #1205, #1215);
#1217=EDGE_CURVE('', #1206, #1213, #1216, .T.);
#1218=ORIENTED_EDGE('', *, *, #1217, .T.);
#1219=CARTESIAN_POINT('', (230.0, -20.0, 40.0));
#1220=VERTEX_POINT('', #1219);
#1221=DIRECTION('', (0.0, 1.0, 0.0));
#1222=VECTOR('', #1221, 20.0);
#1223=LINE('', #1212, #1222);
#1224=EDGE_CURVE('', #1213, #1220, #1223, .T.);
#1225=ORIENTED_EDGE('', *, *, #1224, .T.);
#1226=CARTESIAN_POINT('', (230.0, -20.0, 140.0000000000000028));
#1227=VERTEX_POINT('', #1226);
#1228=DIRECTION('', (0.0, 0.0, 1.0));
#1229=VECTOR('', #1228, 100.0000000000000028);
#1230=LINE('', #1219, #1229);
#1231=EDGE_CURVE('', #1220, #1227, #1230, .T.);
#1232=ORIENTED_EDGE('', *, *, #1231, .T.);
#1233=CARTESIAN_POINT('', (230.0, -40.0, 140.0));
#1234=VERTEX_POINT('', #1233);
#1235=DIRECTION('', (0.0, -1.0, 0.0));
#1236=VECTOR('', #1235, 20.0);
#1237=LINE('', #1226, #1236);

```

```

#1238=EDGE_CURVE('',#1227,#1234,#1237,.T.);
#1239=ORIENTED_EDGE('',*,*,#1238,.T.);
#1240=CARTESIAN_POINT('',(230.0,-40.0,175.0));
#1241=VERTEX_POINT('',#1240);
#1242=DIRECTION('',(0.0,0.0,1.0));
#1243=VECTOR('',#1242,35.0);
#1244=LINE('',#1233,#1243);
#1245=EDGE_CURVE('',#1234,#1241,#1244,.T.);
#1246=ORIENTED_EDGE('',*,*,#1245,.T.);
#1247=DIRECTION('',(0.0,1.0,0.0));
#1248=VECTOR('',#1247,40.0);
#1249=LINE('',#1240,#1248);
#1250=EDGE_CURVE('',#1241,#768,#1249,.T.);
#1251=ORIENTED_EDGE('',*,*,#1250,.T.);
#1252=ORIENTED_EDGE('',*,*,#772,.F.);
#1253=EDGE_LOOP('',(#1191,#1198,#1203,#1204,#1211,#1218,#1225,#1232,#1239,#1246,#1251,#1252));
#1254=FACE_BOUND('',#1253,.F.);
#1255=CARTESIAN_POINT('',(230.0,0.0,0.0));
#1256=DIRECTION('',(-1.0,0.0,0.0));
#1257=DIRECTION('',(0.0,-1.0,0.0));
#1258=AXIS2_PLACEMENT_3D('',#1255,#1256,#1257);
#1259=PLANE('',#1258);
#1260=ADVANCED_FACE('',(#1254),#1259,.F.);
#1261=SHAPE_ASPECT('featured shape: face 11',$,#38,.T.);
#1262=PROPERTY_DEFINITION('',$,#1261);
#1263=FACE_SHAPE_REPRESENTATION('',(#1260),#29);
#1264=PROPERTY_DEFINITION_REPRESENTATION(#1262,#1263);
#1265=CARTESIAN_POINT('',(240.0,-20.0,50.0));
#1266=VERTEX_POINT('',#1265);
#1267=CARTESIAN_POINT('',(240.0,-20.0,40.0));
#1268=DIRECTION('',(0.0,1.0,0.0));
#1269=DIRECTION('',(-1.0,0.0,0.0));
#1270=AXIS2_PLACEMENT_3D('',#1267,#1268,#1269);
#1271=CIRCLE('',#1270,10.0);
#1272=EDGE_CURVE('',#1220,#1266,#1271,.T.);
#1273=ORIENTED_EDGE('',*,*,#1272,.T.);
#1274=CARTESIAN_POINT('',(255.0,-20.0,50.0));
#1275=VERTEX_POINT('',#1274);
#1276=DIRECTION('',(-1.0,0.0,0.0));
#1277=VECTOR('',#1276,15.0);
#1278=LINE('',#1274,#1277);
#1279=EDGE_CURVE('',#1275,#1266,#1278,.T.);
#1280=ORIENTED_EDGE('',*,*,#1279,.F.);
#1281=CARTESIAN_POINT('',(255.0,-20.0,55.0));
#1282=DIRECTION('',(0.0,1.0,0.0));
#1283=DIRECTION('',(1.0,0.0,0.0));
#1284=AXIS2_PLACEMENT_3D('',#1281,#1282,#1283);
#1285=CIRCLE('',#1284,5.0);
#1286=EDGE_CURVE('',#615,#1275,#1285,.T.);
#1287=ORIENTED_EDGE('',*,*,#1286,.F.);
#1288=ORIENTED_EDGE('',*,*,#638,.T.);
#1289=CARTESIAN_POINT('',(255.0,-20.0,130.0));

```



```

#1290=VERTEX_POINT('',#1289);
#1291=CARTESIAN_POINT('',(255.0,-20.0,125.0));
#1292=DIRECTION('',(0.0,-1.0,0.0));
#1293=DIRECTION('',(1.0,0.0,0.0));
#1294=AXIS2_PLACEMENT_3D('',#1291,#1292,#1293);
#1295=CIRCLE('',#1294,5.0);
#1296=EDGE_CURVE('',#629,#1290,#1295,.T.);
#1297=ORIENTED_EDGE('',*,*,#1296,.T.);
#1298=CARTESIAN_POINT('',(240.0,-20.0,130.0));
#1299=VERTEX_POINT('',#1298);
#1300=DIRECTION('',(-1.0,0.0,0.0));
#1301=VECTOR('',#1300,15.0);
#1302=LINE('',#1289,#1301);
#1303=EDGE_CURVE('',#1290,#1299,#1302,.T.);
#1304=ORIENTED_EDGE('',*,*,#1303,.T.);
#1305=CARTESIAN_POINT('',(240.0,-20.0,140.0));
#1306=DIRECTION('',(0.0,-1.0,0.0));
#1307=DIRECTION('',(-1.0,0.0,2.842171E-15));
#1308=AXIS2_PLACEMENT_3D('',#1305,#1306,#1307);
#1309=CIRCLE('',#1308,10.0);
#1310=EDGE_CURVE('',#1227,#1299,#1309,.T.);
#1311=ORIENTED_EDGE('',*,*,#1310,.F.);
#1312=ORIENTED_EDGE('',*,*,#1231,.F.);
#1313=EDGE_LOOP('',(#1273,#1280,#1287,#1288,#1297,#1304,#1311,#1312));
#1314=FACE_BOUND('',#1313,.F.);
#1315=CARTESIAN_POINT('',(250.0,-20.0,110.0));
#1316=VERTEX_POINT('',#1315);
#1317=CARTESIAN_POINT('',(240.0,-20.0,110.0));
#1318=VERTEX_POINT('',#1317);
#1319=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1320=DIRECTION('',(0.0,-1.0,0.0));
#1321=DIRECTION('',(1.0,0.0,0.0));
#1322=AXIS2_PLACEMENT_3D('',#1319,#1320,#1321);
#1323=CIRCLE('',#1322,5.0);
#1324=EDGE_CURVE('',#1316,#1318,#1323,.T.);
#1325=ORIENTED_EDGE('',*,*,#1324,.F.);
#1326=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#1327=DIRECTION('',(0.0,-1.0,0.0));
#1328=DIRECTION('',(-1.0,0.0,0.0));
#1329=AXIS2_PLACEMENT_3D('',#1326,#1327,#1328);
#1330=CIRCLE('',#1329,5.0);
#1331=EDGE_CURVE('',#1318,#1316,#1330,.T.);
#1332=ORIENTED_EDGE('',*,*,#1331,.F.);
#1333=EDGE_LOOP('',(#1325,#1332));
#1334=FACE_BOUND('',#1333,.F.);
#1335=CARTESIAN_POINT('',(250.0,-20.0,70.0));
#1336=VERTEX_POINT('',#1335);
#1337=CARTESIAN_POINT('',(240.0,-20.0,70.0));
#1338=VERTEX_POINT('',#1337);
#1339=CARTESIAN_POINT('',(245.0,-20.0,70.0));
#1340=DIRECTION('',(0.0,-1.0,0.0));
#1341=DIRECTION('',(1.0,0.0,0.0));
#1342=AXIS2_PLACEMENT_3D('',#1339,#1340,#1341);

```

```

#1343=CIRCLE('', #1342, 5.0);
#1344=EDGE_CURVE('', #1336, #1338, #1343, .T.);
#1345=ORIENTED_EDGE('', *, *, #1344, .F.);
#1346=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#1347=DIRECTION('', (0.0, -1.0, 0.0));
#1348=DIRECTION('', (-1.0, 0.0, 0.0));
#1349=AXIS2_PLACEMENT_3D('', #1346, #1347, #1348);
#1350=CIRCLE('', #1349, 5.0);
#1351=EDGE_CURVE('', #1338, #1336, #1350, .T.);
#1352=ORIENTED_EDGE('', *, *, #1351, .F.);
#1353=EDGE_LOOP('', (#1345, #1352));
#1354=FACE_BOUND('', #1353, .F.);
#1355=CARTESIAN_POINT('', (230.0, -20.0, 0.0));
#1356=DIRECTION('', (0.0, -1.0, 0.0));
#1357=DIRECTION('', (1.0, 0.0, 0.0));
#1358=AXIS2_PLACEMENT_3D('', #1355, #1356, #1357);
#1359=PLANE('', #1358);
#1360=ADVANCED_FACE('', (#1314, #1334, #1354), #1359, .F.);
#1361=SHAPE_ASPECT('featured shape: face 12', $, #38, .T.);
#1362=PROPERTY_DEFINITION('', $, #1361);
#1363=FACE_SHAPE_REPRESENTATION('', (#1360), #29);
#1364=PROPERTY_DEFINITION_REPRESENTATION(#1362, #1363);
#1365=ORIENTED_EDGE('', *, *, #46, .T.);
#1366=CARTESIAN_POINT('', (5.0, -40.0, 5.0));
#1367=DIRECTION('', (0.0, 1.0, 0.0));
#1368=DIRECTION('', (0.0, 0.0, -1.0));
#1369=AXIS2_PLACEMENT_3D('', #1366, #1367, #1368);
#1370=CIRCLE('', #1369, 5.0);
#1371=EDGE_CURVE('', #42, #185, #1370, .T.);
#1372=ORIENTED_EDGE('', *, *, #1371, .T.);
#1373=ORIENTED_EDGE('', *, *, #189, .T.);
#1374=CARTESIAN_POINT('', (5.0, -40.0, 175.0));
#1375=DIRECTION('', (0.0, -1.0, 0.0));
#1376=DIRECTION('', (0.0, 0.0, 1.0));
#1377=AXIS2_PLACEMENT_3D('', #1374, #1375, #1376);
#1378=CIRCLE('', #1377, 5.0);
#1379=EDGE_CURVE('', #110, #178, #1378, .T.);
#1380=ORIENTED_EDGE('', *, *, #1379, .F.);
#1381=ORIENTED_EDGE('', *, *, #114, .T.);
#1382=CARTESIAN_POINT('', (225.0, -40.0, 175.0));
#1383=DIRECTION('', (0.0, -1.0, 0.0));
#1384=DIRECTION('', (1.0, 0.0, 0.0));
#1385=AXIS2_PLACEMENT_3D('', #1382, #1383, #1384);
#1386=CIRCLE('', #1385, 5.0);
#1387=EDGE_CURVE('', #1241, #103, #1386, .T.);
#1388=ORIENTED_EDGE('', *, *, #1387, .F.);
#1389=ORIENTED_EDGE('', *, *, #1245, .F.);
#1390=CARTESIAN_POINT('', (240.0, -40.0, 130.0));
#1391=VERTEX_POINT('', #1390);
#1392=CARTESIAN_POINT('', (240.0, -40.0, 140.0));
#1393=DIRECTION('', (0.0, -1.0, 0.0));
#1394=DIRECTION('', (-1.0, 0.0, 5.684342E-15));
#1395=AXIS2_PLACEMENT_3D('', #1392, #1393, #1394);

```

```

#1396=CIRCLE('', #1395, 10.0);
#1397=EDGE_CURVE('', #1234, #1391, #1396, .T.);
#1398=ORIENTED_EDGE('', *, *, #1397, .T.);
#1399=CARTESIAN_POINT('', (255.0, -40.0, 130.0));
#1400=VERTEX_POINT('', #1399);
#1401=DIRECTION('', (-1.0, 0.0, 0.0));
#1402=VECTOR('', #1401, 15.0);
#1403=LINE('', #1399, #1402);
#1404=EDGE_CURVE('', #1400, #1391, #1403, .T.);
#1405=ORIENTED_EDGE('', *, *, #1404, .F.);
#1406=CARTESIAN_POINT('', (255.0, -40.0, 125.0));
#1407=DIRECTION('', (0.0, -1.0, 0.0));
#1408=DIRECTION('', (1.0, 0.0, 0.0));
#1409=AXIS2_PLACEMENT_3D('', #1406, #1407, #1408);
#1410=CIRCLE('', #1409, 5.0);
#1411=EDGE_CURVE('', #622, #1400, #1410, .T.);
#1412=ORIENTED_EDGE('', *, *, #1411, .F.);
#1413=ORIENTED_EDGE('', *, *, #626, .F.);
#1414=CARTESIAN_POINT('', (255.0, -40.0, 50.0));
#1415=VERTEX_POINT('', #1414);
#1416=CARTESIAN_POINT('', (255.0, -40.0, 55.0));
#1417=DIRECTION('', (0.0, 1.0, 0.0));
#1418=DIRECTION('', (1.0, 0.0, 0.0));
#1419=AXIS2_PLACEMENT_3D('', #1416, #1417, #1418);
#1420=CIRCLE('', #1419, 5.0);
#1421=EDGE_CURVE('', #613, #1415, #1420, .T.);
#1422=ORIENTED_EDGE('', *, *, #1421, .T.);
#1423=CARTESIAN_POINT('', (240.0, -40.0, 50.0));
#1424=VERTEX_POINT('', #1423);
#1425=DIRECTION('', (-1.0, 0.0, 0.0));
#1426=VECTOR('', #1425, 15.0);
#1427=LINE('', #1414, #1426);
#1428=EDGE_CURVE('', #1415, #1424, #1427, .T.);
#1429=ORIENTED_EDGE('', *, *, #1428, .T.);
#1430=CARTESIAN_POINT('', (240.0, -40.0, 40.0));
#1431=DIRECTION('', (0.0, 1.0, 0.0));
#1432=DIRECTION('', (-1.0, 0.0, -5.684342E-15));
#1433=AXIS2_PLACEMENT_3D('', #1430, #1431, #1432);
#1434=CIRCLE('', #1433, 10.0);
#1435=EDGE_CURVE('', #1213, #1424, #1434, .T.);
#1436=ORIENTED_EDGE('', *, *, #1435, .F.);
#1437=ORIENTED_EDGE('', *, *, #1217, .F.);
#1438=CARTESIAN_POINT('', (225.0, -40.0, 5.0));
#1439=DIRECTION('', (0.0, 1.0, 0.0));
#1440=DIRECTION('', (1.0, 0.0, 0.0));
#1441=AXIS2_PLACEMENT_3D('', #1438, #1439, #1440);
#1442=CIRCLE('', #1441, 5.0);
#1443=EDGE_CURVE('', #1206, #40, #1442, .T.);
#1444=ORIENTED_EDGE('', *, *, #1443, .T.);
#1445=EDGE_LOOP('', (#1365, #1372, #1373, #1380, #1381, #1388, #1389, #1398, #1405, #1412, #1413, #1422, #1429, #1436, #1437, #1444));
#1446=FACE_BOUND('', #1445, .F.);
#1447=CARTESIAN_POINT('', (45.0, -40.0, 2.0));

```

```

#1448=VERTEX_POINT('', #1447);
#1449=CARTESIAN_POINT('', (7.0, -40.0, 2.0));
#1450=VERTEX_POINT('', #1449);
#1451=DIRECTION('', (-1.0, 0.0, 0.0));
#1452=VECTOR('', #1451, 38.0);
#1453=LINE('', #1447, #1452);
#1454=EDGE_CURVE('', #1448, #1450, #1453, .T.);
#1455=ORIENTED_EDGE('', *, *, #1454, .F.);
#1456=CARTESIAN_POINT('', (50.0, -40.0, 7.0));
#1457=VERTEX_POINT('', #1456);
#1458=CARTESIAN_POINT('', (45.0, -40.0, 7.0));
#1459=DIRECTION('', (0.0, -1.0, 0.0));
#1460=DIRECTION('', (0.0, 0.0, -1.0));
#1461=AXIS2_PLACEMENT_3D('', #1458, #1459, #1460);
#1462=CIRCLE('', #1461, 5.0);
#1463=EDGE_CURVE('', #1448, #1457, #1462, .T.);
#1464=ORIENTED_EDGE('', *, *, #1463, .T.);
#1465=CARTESIAN_POINT('', (50.0, -40.0, 45.0));
#1466=VERTEX_POINT('', #1465);
#1467=DIRECTION('', (0.0, 0.0, -1.0));
#1468=VECTOR('', #1467, 38.0);
#1469=LINE('', #1465, #1468);
#1470=EDGE_CURVE('', #1466, #1457, #1469, .T.);
#1471=ORIENTED_EDGE('', *, *, #1470, .F.);
#1472=CARTESIAN_POINT('', (45.0, -40.0, 50.0));
#1473=VERTEX_POINT('', #1472);
#1474=CARTESIAN_POINT('', (45.0, -40.0, 45.000000000000007));
#1475=DIRECTION('', (0.0, 1.0, 0.0));
#1476=DIRECTION('', (0.0, 0.0, 1.0));
#1477=AXIS2_PLACEMENT_3D('', #1474, #1475, #1476);
#1478=CIRCLE('', #1477, 5.0);
#1479=EDGE_CURVE('', #1473, #1466, #1478, .T.);
#1480=ORIENTED_EDGE('', *, *, #1479, .F.);
#1481=CARTESIAN_POINT('', (7.0, -40.0, 50.0));
#1482=VERTEX_POINT('', #1481);
#1483=DIRECTION('', (1.0, 0.0, 0.0));
#1484=VECTOR('', #1483, 38.0);
#1485=LINE('', #1481, #1484);
#1486=EDGE_CURVE('', #1482, #1473, #1485, .T.);
#1487=ORIENTED_EDGE('', *, *, #1486, .F.);
#1488=CARTESIAN_POINT('', (2.0, -40.0, 45.0));
#1489=VERTEX_POINT('', #1488);
#1490=CARTESIAN_POINT('', (7.0, -40.0, 45.000000000000007));
#1491=DIRECTION('', (0.0, 1.0, 0.0));
#1492=DIRECTION('', (-1.0, 0.0, -1.421085E-15));
#1493=AXIS2_PLACEMENT_3D('', #1490, #1491, #1492);
#1494=CIRCLE('', #1493, 5.0);
#1495=EDGE_CURVE('', #1489, #1482, #1494, .T.);
#1496=ORIENTED_EDGE('', *, *, #1495, .F.);
#1497=CARTESIAN_POINT('', (2.0, -40.0, 7.0));
#1498=VERTEX_POINT('', #1497);
#1499=DIRECTION('', (0.0, 0.0, 1.0));
#1500=VECTOR('', #1499, 38.0);

```

```

#1501=LINE('',#1497,#1500);
#1502=EDGE_CURVE('',#1498,#1489,#1501,.T.);
#1503=ORIENTED_EDGE('',*,*,#1502,.F.);
#1504=CARTESIAN_POINT('',(7.0,-40.0,7.0));
#1505=DIRECTION('',(0.0,-1.0,0.0));
#1506=DIRECTION('',(-1.0,0.0,0.0));
#1507=AXIS2_PLACEMENT_3D('',#1504,#1505,#1506);
#1508=CIRCLE('',#1507,5.0);
#1509=EDGE_CURVE('',#1498,#1450,#1508,.T.);
#1510=ORIENTED_EDGE('',*,*,#1509,.T.);
#1511=EDGE_LOOP('',(#1455,#1464,#1471,#1480,#1487,#1496,#1503,#1510));
#1512=FACE_BOUND('',#1511,.F.);
#1513=CARTESIAN_POINT('',(228.0,-40.0,45.0));
#1514=VERTEX_POINT('',#1513);
#1515=CARTESIAN_POINT('',(228.0,-40.0,7.0));
#1516=VERTEX_POINT('',#1515);
#1517=DIRECTION('',(0.0,0.0,-1.0));
#1518=VECTOR('',#1517,38.0);
#1519=LINE('',#1513,#1518);
#1520=EDGE_CURVE('',#1514,#1516,#1519,.T.);
#1521=ORIENTED_EDGE('',*,*,#1520,.F.);
#1522=CARTESIAN_POINT('',(223.0,-40.0,50.0));
#1523=VERTEX_POINT('',#1522);
#1524=CARTESIAN_POINT('',(223.0,-40.0,45.0));
#1525=DIRECTION('',(0.0,-1.0,0.0));
#1526=DIRECTION('',(1.0,0.0,0.0));
#1527=AXIS2_PLACEMENT_3D('',#1524,#1525,#1526);
#1528=CIRCLE('',#1527,5.0);
#1529=EDGE_CURVE('',#1514,#1523,#1528,.T.);
#1530=ORIENTED_EDGE('',*,*,#1529,.T.);
#1531=CARTESIAN_POINT('',(185.000000000000028,-40.0,50.0));
#1532=VERTEX_POINT('',#1531);
#1533=DIRECTION('',(1.0,0.0,0.0));
#1534=VECTOR('',#1533,37.999999999999972);
#1535=LINE('',#1531,#1534);
#1536=EDGE_CURVE('',#1532,#1523,#1535,.T.);
#1537=ORIENTED_EDGE('',*,*,#1536,.F.);
#1538=CARTESIAN_POINT('',(180.000000000000028,-40.0,45.0));
#1539=VERTEX_POINT('',#1538);
#1540=CARTESIAN_POINT('',(185.000000000000028,-40.0,45.0));
#1541=DIRECTION('',(0.0,1.0,0.0));
#1542=DIRECTION('',(-1.0,0.0,0.0));
#1543=AXIS2_PLACEMENT_3D('',#1540,#1541,#1542);
#1544=CIRCLE('',#1543,5.0);
#1545=EDGE_CURVE('',#1539,#1532,#1544,.T.);
#1546=ORIENTED_EDGE('',*,*,#1545,.F.);
#1547=CARTESIAN_POINT('',(180.000000000000028,-40.0,7.0));
#1548=VERTEX_POINT('',#1547);
#1549=DIRECTION('',(0.0,0.0,1.0));
#1550=VECTOR('',#1549,38.0);
#1551=LINE('',#1547,#1550);
#1552=EDGE_CURVE('',#1548,#1539,#1551,.T.);
#1553=ORIENTED_EDGE('',*,*,#1552,.F.);

```

```

#1554=CARTESIAN_POINT('', (185.000000000000028, -40.0, 2.0));
#1555=VERTEX_POINT('', #1554);
#1556=CARTESIAN_POINT('', (185.000000000000028, -40.0, 7.0));
#1557=DIRECTION('', (0.0, 1.0, 0.0));
#1558=DIRECTION('', (0.0, 0.0, -1.0));
#1559=AXIS2_PLACEMENT_3D('', #1556, #1557, #1558);
#1560=CIRCLE('', #1559, 5.0);
#1561=EDGE_CURVE('', #1555, #1548, #1560, .T.);
#1562=ORIENTED_EDGE('', *, *, #1561, .F.);
#1563=CARTESIAN_POINT('', (223.0, -40.0, 2.0));
#1564=VERTEX_POINT('', #1563);
#1565=DIRECTION('', (-1.0, 0.0, 0.0));
#1566=VECTOR('', #1565, 37.999999999999972);
#1567=LINE('', #1563, #1566);
#1568=EDGE_CURVE('', #1564, #1555, #1567, .T.);
#1569=ORIENTED_EDGE('', *, *, #1568, .F.);
#1570=CARTESIAN_POINT('', (223.0, -40.0, 7.0));
#1571=DIRECTION('', (0.0, 1.0, 0.0));
#1572=DIRECTION('', (1.0, 0.0, 0.0));
#1573=AXIS2_PLACEMENT_3D('', #1570, #1571, #1572);
#1574=CIRCLE('', #1573, 5.0);
#1575=EDGE_CURVE('', #1516, #1564, #1574, .T.);
#1576=ORIENTED_EDGE('', *, *, #1575, .F.);
#1577=EDGE_LOOP('', (#1521, #1530, #1537, #1546, #1553, #1562, #1569, #1576));
#1578=FACE_BOUND('', #1577, .F.);
#1579=CARTESIAN_POINT('', (184.999999999999972, -40.0, 178.0));
#1580=VERTEX_POINT('', #1579);
#1581=CARTESIAN_POINT('', (223.0, -40.0, 178.0));
#1582=VERTEX_POINT('', #1581);
#1583=DIRECTION('', (1.0, 0.0, 0.0));
#1584=VECTOR('', #1583, 38.000000000000028);
#1585=LINE('', #1579, #1584);
#1586=EDGE_CURVE('', #1580, #1582, #1585, .T.);
#1587=ORIENTED_EDGE('', *, *, #1586, .F.);
#1588=CARTESIAN_POINT('', (179.999999999999972, -40.0, 173.0));
#1589=VERTEX_POINT('', #1588);
#1590=CARTESIAN_POINT('', (184.999999999999972, -40.0, 173.0));
#1591=DIRECTION('', (0.0, -1.0, 0.0));
#1592=DIRECTION('', (0.0, 0.0, 1.0));
#1593=AXIS2_PLACEMENT_3D('', #1590, #1591, #1592);
#1594=CIRCLE('', #1593, 5.0);
#1595=EDGE_CURVE('', #1580, #1589, #1594, .T.);
#1596=ORIENTED_EDGE('', *, *, #1595, .T.);
#1597=CARTESIAN_POINT('', (179.999999999999972, -40.0, 135.000000000000028));
#1598=VERTEX_POINT('', #1597);
#1599=DIRECTION('', (0.0, 0.0, 1.0));
#1600=VECTOR('', #1599, 37.999999999999972);
#1601=LINE('', #1597, #1600);
#1602=EDGE_CURVE('', #1598, #1589, #1601, .T.);
#1603=ORIENTED_EDGE('', *, *, #1602, .F.);
#1604=CARTESIAN_POINT('', (184.999999999999972, -40.0, 130.000000000000028));
#1605=VERTEX_POINT('', #1604);
#1606=CARTESIAN_POINT('', (184.999999999999972, -40.0, 135.000000000000028));

```

```

#1607=DIRECTION('', (0.0,1.0,0.0));
#1608=DIRECTION('', (0.0,0.0,-1.0));
#1609=AXIS2_PLACEMENT_3D('', #1606, #1607, #1608);
#1610=CIRCLE('', #1609, 5.0);
#1611=EDGE_CURVE('', #1605, #1598, #1610, .T.);
#1612=ORIENTED_EDGE('', *, *, #1611, .F.);
#1613=CARTESIAN_POINT('', (223.0, -40.0, 130.0000000000000028));
#1614=VERTEX_POINT('', #1613);
#1615=DIRECTION('', (-1.0,0.0,0.0));
#1616=VECTOR('', #1615, 38.0000000000000028);
#1617=LINE('', #1613, #1616);
#1618=EDGE_CURVE('', #1614, #1605, #1617, .T.);
#1619=ORIENTED_EDGE('', *, *, #1618, .F.);
#1620=CARTESIAN_POINT('', (228.0, -40.0, 135.0000000000000028));
#1621=VERTEX_POINT('', #1620);
#1622=CARTESIAN_POINT('', (223.0, -40.0, 135.0000000000000028));
#1623=DIRECTION('', (0.0,1.0,0.0));
#1624=DIRECTION('', (1.0,0.0,0.0));
#1625=AXIS2_PLACEMENT_3D('', #1622, #1623, #1624);
#1626=CIRCLE('', #1625, 5.0);
#1627=EDGE_CURVE('', #1621, #1614, #1626, .T.);
#1628=ORIENTED_EDGE('', *, *, #1627, .F.);
#1629=CARTESIAN_POINT('', (228.0, -40.0, 173.0));
#1630=VERTEX_POINT('', #1629);
#1631=DIRECTION('', (0.0,0.0,-1.0));
#1632=VECTOR('', #1631, 37.999999999999972);
#1633=LINE('', #1629, #1632);
#1634=EDGE_CURVE('', #1630, #1621, #1633, .T.);
#1635=ORIENTED_EDGE('', *, *, #1634, .F.);
#1636=CARTESIAN_POINT('', (223.0, -40.0, 173.0));
#1637=DIRECTION('', (0.0,-1.0,0.0));
#1638=DIRECTION('', (1.0,0.0,0.0));
#1639=AXIS2_PLACEMENT_3D('', #1636, #1637, #1638);
#1640=CIRCLE('', #1639, 5.0);
#1641=EDGE_CURVE('', #1630, #1582, #1640, .T.);
#1642=ORIENTED_EDGE('', *, *, #1641, .T.);
#1643=EDGE_LOOP('', (#1587, #1596, #1603, #1612, #1619, #1628, #1635, #1642));
#1644=FACE_BOUND('', #1643, .F.);
#1645=CARTESIAN_POINT('', (167.500000000000000, -40.0, 150.0));
#1646=VERTEX_POINT('', #1645);
#1647=CARTESIAN_POINT('', (152.500000000000000, -40.0, 150.0));
#1648=VERTEX_POINT('', #1647);
#1649=CARTESIAN_POINT('', (160.0, -40.0, 150.0));
#1650=DIRECTION('', (0.0,1.0,0.0));
#1651=DIRECTION('', (1.0,0.0,0.0));
#1652=AXIS2_PLACEMENT_3D('', #1649, #1650, #1651);
#1653=CIRCLE('', #1652, 7.500000000000000);
#1654=EDGE_CURVE('', #1646, #1648, #1653, .T.);
#1655=ORIENTED_EDGE('', *, *, #1654, .F.);
#1656=CARTESIAN_POINT('', (160.0, -40.0, 150.0));
#1657=DIRECTION('', (0.0,1.0,0.0));
#1658=DIRECTION('', (-1.0,0.0,0.0));
#1659=AXIS2_PLACEMENT_3D('', #1656, #1657, #1658);

```

```

#1660=CIRCLE('',#1659,7.500000000000000);
#1661=EDGE_CURVE('',#1648,#1646,#1660,.T.);
#1662=ORIENTED_EDGE('',*,*,#1661,.F.);
#1663=EDGE_LOOP('',(#1655,#1662));
#1664=FACE_BOUND('',#1663,.F.);
#1665=CARTESIAN_POINT('',(180.0,-40.0,90.0));
#1666=VERTEX_POINT('',#1665);
#1667=CARTESIAN_POINT('',(140.0,-40.0,90.0));
#1668=VERTEX_POINT('',#1667);
#1669=CARTESIAN_POINT('',(160.0,-40.0,90.0));
#1670=DIRECTION('',(0.0,-1.0,0.0));
#1671=DIRECTION('',(1.0,0.0,0.0));
#1672=AXIS2_PLACEMENT_3D('',#1669,#1670,#1671);
#1673=CIRCLE('',#1672,20.0);
#1674=EDGE_CURVE('',#1666,#1668,#1673,.T.);
#1675=ORIENTED_EDGE('',*,*,#1674,.T.);
#1676=CARTESIAN_POINT('',(160.0,-40.0,90.0));
#1677=DIRECTION('',(0.0,-1.0,0.0));
#1678=DIRECTION('',(-1.0,0.0,0.0));
#1679=AXIS2_PLACEMENT_3D('',#1676,#1677,#1678);
#1680=CIRCLE('',#1679,20.0);
#1681=EDGE_CURVE('',#1668,#1666,#1680,.T.);
#1682=ORIENTED_EDGE('',*,*,#1681,.T.);
#1683=EDGE_LOOP('',(#1675,#1682));
#1684=FACE_BOUND('',#1683,.F.);
#1685=CARTESIAN_POINT('',(250.0,-40.0,110.0));
#1686=VERTEX_POINT('',#1685);
#1687=CARTESIAN_POINT('',(240.0,-40.0,110.0));
#1688=VERTEX_POINT('',#1687);
#1689=CARTESIAN_POINT('',(245.0,-40.0,110.0));
#1690=DIRECTION('',(0.0,-1.0,0.0));
#1691=DIRECTION('',(1.0,0.0,0.0));
#1692=AXIS2_PLACEMENT_3D('',#1689,#1690,#1691);
#1693=CIRCLE('',#1692,5.0);
#1694=EDGE_CURVE('',#1686,#1688,#1693,.T.);
#1695=ORIENTED_EDGE('',*,*,#1694,.T.);
#1696=CARTESIAN_POINT('',(245.0,-40.0,110.0));
#1697=DIRECTION('',(0.0,-1.0,0.0));
#1698=DIRECTION('',(-1.0,0.0,0.0));
#1699=AXIS2_PLACEMENT_3D('',#1696,#1697,#1698);
#1700=CIRCLE('',#1699,5.0);
#1701=EDGE_CURVE('',#1688,#1686,#1700,.T.);
#1702=ORIENTED_EDGE('',*,*,#1701,.T.);
#1703=EDGE_LOOP('',(#1695,#1702));
#1704=FACE_BOUND('',#1703,.F.);
#1705=CARTESIAN_POINT('',(250.0,-40.0,70.0));
#1706=VERTEX_POINT('',#1705);
#1707=CARTESIAN_POINT('',(240.0,-40.0,70.0));
#1708=VERTEX_POINT('',#1707);
#1709=CARTESIAN_POINT('',(245.0,-40.0,70.0));
#1710=DIRECTION('',(0.0,-1.0,0.0));
#1711=DIRECTION('',(1.0,0.0,0.0));
#1712=AXIS2_PLACEMENT_3D('',#1709,#1710,#1711);

```



```

#1713=CIRCLE('', #1712, 5.0);
#1714=EDGE_CURVE('', #1706, #1708, #1713, .T.);
#1715=ORIENTED_EDGE('', *, *, #1714, .T.);
#1716=CARTESIAN_POINT('', (245.0, -40.0, 70.0));
#1717=DIRECTION('', (0.0, -1.0, 0.0));
#1718=DIRECTION('', (-1.0, 0.0, 0.0));
#1719=AXIS2_PLACEMENT_3D('', #1716, #1717, #1718);
#1720=CIRCLE('', #1719, 5.0);
#1721=EDGE_CURVE('', #1708, #1706, #1720, .T.);
#1722=ORIENTED_EDGE('', *, *, #1721, .T.);
#1723=EDGE_LOOP('', (#1715, #1722));
#1724=FACE_BOUND('', #1723, .F.);
#1725=CARTESIAN_POINT('', (2.0, -40.0, 134.999999999999972));
#1726=VERTEX_POINT('', #1725);
#1727=CARTESIAN_POINT('', (2.0, -40.0, 173.0));
#1728=VERTEX_POINT('', #1727);
#1729=DIRECTION('', (0.0, 0.0, 1.0));
#1730=VECTOR('', #1729, 38.0000000000000028);
#1731=LINE('', #1725, #1730);
#1732=EDGE_CURVE('', #1726, #1728, #1731, .T.);
#1733=ORIENTED_EDGE('', *, *, #1732, .F.);
#1734=CARTESIAN_POINT('', (7.0, -40.0, 129.999999999999972));
#1735=VERTEX_POINT('', #1734);
#1736=CARTESIAN_POINT('', (7.0, -40.0, 134.999999999999972));
#1737=DIRECTION('', (0.0, -1.0, 0.0));
#1738=DIRECTION('', (-1.0, 0.0, 0.0));
#1739=AXIS2_PLACEMENT_3D('', #1736, #1737, #1738);
#1740=CIRCLE('', #1739, 5.0);
#1741=EDGE_CURVE('', #1726, #1735, #1740, .T.);
#1742=ORIENTED_EDGE('', *, *, #1741, .T.);
#1743=CARTESIAN_POINT('', (44.999999999999972, -40.0, 129.999999999999972));
#1744=VERTEX_POINT('', #1743);
#1745=DIRECTION('', (-1.0, 0.0, 0.0));
#1746=VECTOR('', #1745, 37.999999999999972);
#1747=LINE('', #1743, #1746);
#1748=EDGE_CURVE('', #1744, #1735, #1747, .T.);
#1749=ORIENTED_EDGE('', *, *, #1748, .F.);
#1750=CARTESIAN_POINT('', (49.999999999999972, -40.0, 134.999999999999972));
#1751=VERTEX_POINT('', #1750);
#1752=CARTESIAN_POINT('', (44.999999999999972, -40.0, 134.999999999999972));
#1753=DIRECTION('', (0.0, 1.0, 0.0));
#1754=DIRECTION('', (1.0, 0.0, 0.0));
#1755=AXIS2_PLACEMENT_3D('', #1752, #1753, #1754);
#1756=CIRCLE('', #1755, 5.0);
#1757=EDGE_CURVE('', #1751, #1744, #1756, .T.);
#1758=ORIENTED_EDGE('', *, *, #1757, .F.);
#1759=CARTESIAN_POINT('', (49.999999999999972, -40.0, 173.0));
#1760=VERTEX_POINT('', #1759);
#1761=DIRECTION('', (0.0, 0.0, -1.0));
#1762=VECTOR('', #1761, 38.0000000000000028);
#1763=LINE('', #1759, #1762);
#1764=EDGE_CURVE('', #1760, #1751, #1763, .T.);
#1765=ORIENTED_EDGE('', *, *, #1764, .F.);

```

```

#1766=CARTESIAN_POINT('', (44.999999999999972, -40.0, 178.0));
#1767=VERTEX_POINT('', #1766);
#1768=CARTESIAN_POINT('', (44.999999999999972, -40.0, 173.0));
#1769=DIRECTION('', (0.0, 1.0, 0.0));
#1770=DIRECTION('', (0.0, 0.0, 1.0));
#1771=AXIS2_PLACEMENT_3D('', #1768, #1769, #1770);
#1772=CIRCLE('', #1771, 5.0);
#1773=EDGE_CURVE('', #1767, #1760, #1772, .T.);
#1774=ORIENTED_EDGE('', *, *, #1773, .F.);
#1775=CARTESIAN_POINT('', (7.0, -40.0, 178.0));
#1776=VERTEX_POINT('', #1775);
#1777=DIRECTION('', (1.0, 0.0, 0.0));
#1778=VECTOR('', #1777, 37.999999999999972);
#1779=LINE('', #1775, #1778);
#1780=EDGE_CURVE('', #1776, #1767, #1779, .T.);
#1781=ORIENTED_EDGE('', *, *, #1780, .F.);
#1782=CARTESIAN_POINT('', (7.0, -40.0, 173.0));
#1783=DIRECTION('', (0.0, 1.0, 0.0));
#1784=DIRECTION('', (-1.0, 0.0, 0.0));
#1785=AXIS2_PLACEMENT_3D('', #1782, #1783, #1784);
#1786=CIRCLE('', #1785, 5.0);
#1787=EDGE_CURVE('', #1728, #1776, #1786, .T.);
#1788=ORIENTED_EDGE('', *, *, #1787, .F.);
#1789=EDGE_LOOP('', (#1733, #1742, #1749, #1758, #1765, #1774, #1781, #1788));
#1790=FACE_BOUND('', #1789, .F.);
#1791=CARTESIAN_POINT('', (230.0, -40.0, 180.0));
#1792=DIRECTION('', (0.0, 1.0, 0.0));
#1793=DIRECTION('', (1.0, 0.0, 0.0));
#1794=AXIS2_PLACEMENT_3D('', #1791, #1792, #1793);
#1795=PLANE('', #1794);
#1796=ADVANCED_FACE('', (#1446, #1512, #1578, #1644, #1664, #1684, #1704, #1724, #1790
), #1795, .F.);
#1797=SHAPE_ASPECT('featured shape: face 13', $, #38, .T.);
#1798=PROPERTY_DEFINITION('', $, #1797);
#1799=FACE_SHAPE_REPRESENTATION('', (#1796), #29);
#1800=PROPERTY_DEFINITION_REPRESENTATION(#1798, #1799);
#1801=ORIENTED_EDGE('', *, *, #1397, .F.);
#1802=ORIENTED_EDGE('', *, *, #1238, .F.);
#1803=ORIENTED_EDGE('', *, *, #1310, .T.);
#1804=DIRECTION('', (0.0, -1.0, 0.0));
#1805=VECTOR('', #1804, 20.0);
#1806=LINE('', #1298, #1805);
#1807=EDGE_CURVE('', #1299, #1391, #1806, .T.);
#1808=ORIENTED_EDGE('', *, *, #1807, .T.);
#1809=EDGE_LOOP('', (#1801, #1802, #1803, #1808));
#1810=FACE_BOUND('', #1809, .F.);
#1811=CARTESIAN_POINT('', (240.0, -10.0, 140.0));
#1812=DIRECTION('', (0.0, -1.0, 0.0));
#1813=DIRECTION('', (0.0, 0.0, -1.0));
#1814=AXIS2_PLACEMENT_3D('', #1811, #1812, #1813);
#1815=CYLINDRICAL_SURFACE('', #1814, 10.0);
#1816=ADVANCED_FACE('', (#1810), #1815, .F.);
#1817=SHAPE_ASPECT('featured shape: face 14', $, #38, .T.);

```

```

#1818=PROPERTY_DEFINITION('', $, #1817);
#1819=FACE_SHAPE_REPRESENTATION('', (#1816), #29);
#1820=PROPERTY_DEFINITION_REPRESENTATION(#1818, #1819);
#1821=DIRECTION('', (0.0, 1.0, 0.0));
#1822=VECTOR('', #1821, 20.0);
#1823=LINE('', #1399, #1822);
#1824=EDGE_CURVE('', #1400, #1290, #1823, .T.);
#1825=ORIENTED_EDGE('', *, *, #1824, .F.);
#1826=ORIENTED_EDGE('', *, *, #1404, .T.);
#1827=ORIENTED_EDGE('', *, *, #1807, .F.);
#1828=ORIENTED_EDGE('', *, *, #1303, .F.);
#1829=EDGE_LOOP('', (#1825, #1826, #1827, #1828));
#1830=FACE_BOUND('', #1829, .F.);
#1831=CARTESIAN_POINT('', (240.0, -10.0, 130.0));
#1832=DIRECTION('', (0.0, 0.0, -1.0));
#1833=DIRECTION('', (1.0, 0.0, 0.0));
#1834=AXIS2_PLACEMENT_3D('', #1831, #1832, #1833);
#1835=PLANE('', #1834);
#1836=ADVANCED_FACE('', (#1830), #1835, .F.);
#1837=SHAPE_ASPECT('featured shape: face 15', $, #38, .T.);
#1838=PROPERTY_DEFINITION('', $, #1837);
#1839=FACE_SHAPE_REPRESENTATION('', (#1836), #29);
#1840=PROPERTY_DEFINITION_REPRESENTATION(#1838, #1839);
#1841=ORIENTED_EDGE('', *, *, #1435, .T.);
#1842=DIRECTION('', (0.0, 1.0, 0.0));
#1843=VECTOR('', #1842, 20.0);
#1844=LINE('', #1423, #1843);
#1845=EDGE_CURVE('', #1424, #1266, #1844, .T.);
#1846=ORIENTED_EDGE('', *, *, #1845, .T.);
#1847=ORIENTED_EDGE('', *, *, #1272, .F.);
#1848=ORIENTED_EDGE('', *, *, #1224, .F.);
#1849=EDGE_LOOP('', (#1841, #1846, #1847, #1848));
#1850=FACE_BOUND('', #1849, .F.);
#1851=CARTESIAN_POINT('', (240.0, -50.0, 40.0));
#1852=DIRECTION('', (0.0, 1.0, 0.0));
#1853=DIRECTION('', (0.0, 0.0, 1.0));
#1854=AXIS2_PLACEMENT_3D('', #1851, #1852, #1853);
#1855=CYLINDRICAL_SURFACE('', #1854, 10.0);
#1856=ADVANCED_FACE('', (#1850), #1855, .F.);
#1857=SHAPE_ASPECT('featured shape: face 16', $, #38, .T.);
#1858=PROPERTY_DEFINITION('', $, #1857);
#1859=FACE_SHAPE_REPRESENTATION('', (#1856), #29);
#1860=PROPERTY_DEFINITION_REPRESENTATION(#1858, #1859);
#1861=DIRECTION('', (0.0, 1.0, 0.0));
#1862=VECTOR('', #1861, 20.0);
#1863=LINE('', #1414, #1862);
#1864=EDGE_CURVE('', #1415, #1275, #1863, .T.);
#1865=ORIENTED_EDGE('', *, *, #1864, .T.);
#1866=ORIENTED_EDGE('', *, *, #1279, .T.);
#1867=ORIENTED_EDGE('', *, *, #1845, .F.);
#1868=ORIENTED_EDGE('', *, *, #1428, .F.);
#1869=EDGE_LOOP('', (#1865, #1866, #1867, #1868));
#1870=FACE_BOUND('', #1869, .F.);

```

ISO 10303-223:2008 (E)

```

#1871=CARTESIAN_POINT('', (240.0, -50.0, 50.0));
#1872=DIRECTION('', (0.0, 0.0, 1.0));
#1873=DIRECTION('', (1.0, 0.0, 0.0));
#1874=AXIS2_PLACEMENT_3D('', #1871, #1872, #1873);
#1875=PLANE('', #1874);
#1876=ADVANCED_FACE('', (#1870), #1875, .F.);
#1877=SHAPE_ASPECT('featured shape: face 17', $, #38, .T.);
#1878=PROPERTY_DEFINITION('', $, #1877);
#1879=FACE_SHAPE_REPRESENTATION('', (#1876), #29);
#1880=PROPERTY_DEFINITION_REPRESENTATION(#1878, #1879);
#1881=ORIENTED_EDGE('', *, *, #1324, .T.);
#1882=DIRECTION('', (0.0, -1.0, 0.0));
#1883=VECTOR('', #1882, 20.0);
#1884=LINE('', #1317, #1883);
#1885=EDGE_CURVE('', #1318, #1688, #1884, .T.);
#1886=ORIENTED_EDGE('', *, *, #1885, .T.);
#1887=ORIENTED_EDGE('', *, *, #1694, .F.);
#1888=DIRECTION('', (0.0, -1.0, 0.0));
#1889=VECTOR('', #1888, 20.0);
#1890=LINE('', #1315, #1889);
#1891=EDGE_CURVE('', #1316, #1686, #1890, .T.);
#1892=ORIENTED_EDGE('', *, *, #1891, .F.);
#1893=EDGE_LOOP('', (#1881, #1886, #1887, #1892));
#1894=FACE_BOUND('', #1893, .F.);
#1895=CARTESIAN_POINT('', (245.0, -20.0, 110.0));
#1896=DIRECTION('', (0.0, -1.0, 0.0));
#1897=DIRECTION('', (1.0, 0.0, 0.0));
#1898=AXIS2_PLACEMENT_3D('', #1895, #1896, #1897);
#1899=CYLINDRICAL_SURFACE('', #1898, 5.0);
#1900=ADVANCED_FACE('', (#1894), #1899, .F.);
#1901=SHAPE_ASPECT('featured shape: face 18', $, #38, .T.);
#1902=PROPERTY_DEFINITION('', $, #1901);
#1903=FACE_SHAPE_REPRESENTATION('', (#1900), #29);
#1904=PROPERTY_DEFINITION_REPRESENTATION(#1902, #1903);
#1905=ORIENTED_EDGE('', *, *, #1331, .T.);
#1906=ORIENTED_EDGE('', *, *, #1891, .T.);
#1907=ORIENTED_EDGE('', *, *, #1701, .F.);
#1908=ORIENTED_EDGE('', *, *, #1885, .F.);
#1909=EDGE_LOOP('', (#1905, #1906, #1907, #1908));
#1910=FACE_BOUND('', #1909, .F.);
#1911=CARTESIAN_POINT('', (245.0, -20.0, 110.0));
#1912=DIRECTION('', (0.0, -1.0, 0.0));
#1913=DIRECTION('', (1.0, 0.0, 0.0));
#1914=AXIS2_PLACEMENT_3D('', #1911, #1912, #1913);
#1915=CYLINDRICAL_SURFACE('', #1914, 5.0);
#1916=ADVANCED_FACE('', (#1910), #1915, .F.);
#1917=SHAPE_ASPECT('featured shape: face 19', $, #38, .T.);
#1918=PROPERTY_DEFINITION('', $, #1917);
#1919=FACE_SHAPE_REPRESENTATION('', (#1916), #29);
#1920=PROPERTY_DEFINITION_REPRESENTATION(#1918, #1919);
#1921=ORIENTED_EDGE('', *, *, #1344, .T.);
#1922=DIRECTION('', (0.0, -1.0, 0.0));
#1923=VECTOR('', #1922, 20.0);

```

```

#1924=LINE('', #1337, #1923);
#1925=EDGE_CURVE('', #1338, #1708, #1924, .T.);
#1926=ORIENTED_EDGE('', *, *, #1925, .T.);
#1927=ORIENTED_EDGE('', *, *, #1714, .F.);
#1928=DIRECTION('', (0.0, -1.0, 0.0));
#1929=VECTOR('', #1928, 20.0);
#1930=LINE('', #1335, #1929);
#1931=EDGE_CURVE('', #1336, #1706, #1930, .T.);
#1932=ORIENTED_EDGE('', *, *, #1931, .F.);
#1933=EDGE_LOOP('', (#1921, #1926, #1927, #1932));
#1934=FACE_BOUND('', #1933, .F.);
#1935=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#1936=DIRECTION('', (0.0, -1.0, 0.0));
#1937=DIRECTION('', (1.0, 0.0, 0.0));
#1938=AXIS2_PLACEMENT_3D('', #1935, #1936, #1937);
#1939=CYLINDRICAL_SURFACE('', #1938, 5.0);
#1940=ADVANCED_FACE('', (#1934), #1939, .F.);
#1941=SHAPE_ASPECT('featured shape: face 20', $, #38, .T.);
#1942=PROPERTY_DEFINITION('', $, #1941);
#1943=FACE_SHAPE_REPRESENTATION('', (#1940), #29);
#1944=PROPERTY_DEFINITION_REPRESENTATION(#1942, #1943);
#1945=ORIENTED_EDGE('', *, *, #1351, .T.);
#1946=ORIENTED_EDGE('', *, *, #1931, .T.);
#1947=ORIENTED_EDGE('', *, *, #1721, .F.);
#1948=ORIENTED_EDGE('', *, *, #1925, .F.);
#1949=EDGE_LOOP('', (#1945, #1946, #1947, #1948));
#1950=FACE_BOUND('', #1949, .F.);
#1951=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#1952=DIRECTION('', (0.0, -1.0, 0.0));
#1953=DIRECTION('', (1.0, 0.0, 0.0));
#1954=AXIS2_PLACEMENT_3D('', #1951, #1952, #1953);
#1955=CYLINDRICAL_SURFACE('', #1954, 5.0);
#1956=ADVANCED_FACE('', (#1950), #1955, .F.);
#1957=SHAPE_ASPECT('featured shape: face 21', $, #38, .T.);
#1958=PROPERTY_DEFINITION('', $, #1957);
#1959=FACE_SHAPE_REPRESENTATION('', (#1956), #29);
#1960=PROPERTY_DEFINITION_REPRESENTATION(#1958, #1959);
#1961=CARTESIAN_POINT('', (180.0, -18.250000000000000, 90.000000000000014));
#1962=VERTEX_POINT('', #1961);
#1963=CARTESIAN_POINT('', (140.0, -18.250000000000000, 90.0));
#1964=VERTEX_POINT('', #1963);
#1965=CARTESIAN_POINT('', (160.0, -18.250000000000000, 90.0));
#1966=DIRECTION('', (0.0, -1.0, 0.0));
#1967=DIRECTION('', (1.0, 0.0, 7.105427E-16));
#1968=AXIS2_PLACEMENT_3D('', #1965, #1966, #1967);
#1969=CIRCLE('', #1968, 20.0);
#1970=EDGE_CURVE('', #1962, #1964, #1969, .T.);
#1971=ORIENTED_EDGE('', *, *, #1970, .F.);
#1972=DIRECTION('', (0.0, -1.0, 7.786770E-16));
#1973=VECTOR('', #1972, 18.250000000000000);
#1974=LINE('', #784, #1973);
#1975=EDGE_CURVE('', #785, #1962, #1974, .T.);
#1976=ORIENTED_EDGE('', *, *, #1975, .F.);

```

ISO 10303-223:2008 (E)

```

#1977=ORIENTED_EDGE('',*,*,#793,.T.);
#1978=DIRECTION('',(0.0,-1.0,0.0));
#1979=VECTOR('',#1978,18.250000000000000);
#1980=LINE('',#786,#1979);
#1981=EDGE_CURVE('',#787,#1964,#1980,.T.);
#1982=ORIENTED_EDGE('',*,*,#1981,.T.);
#1983=EDGE_LOOP('',(#1971,#1976,#1977,#1982));
#1984=FACE_BOUND('',#1983,.F.);
#1985=CARTESIAN_POINT('',(160.0,0.0,90.0));
#1986=DIRECTION('',(0.0,-1.0,0.0));
#1987=DIRECTION('',(1.0,0.0,0.0));
#1988=AXIS2_PLACEMENT_3D('',#1985,#1986,#1987);
#1989=CYLINDRICAL_SURFACE('',#1988,20.0);
#1990=ADVANCED_FACE('',(#1984),#1989,.F.);
#1991=SHAPE_ASPECT('featured shape: face 22',$,#38,.T.);
#1992=PROPERTY_DEFINITION('',$,#1991);
#1993=FACE_SHAPE_REPRESENTATION('',(#1990),#29);
#1994=PROPERTY_DEFINITION_REPRESENTATION(#1992,#1993);
#1995=ORIENTED_EDGE('',*,*,#1674,.F.);
#1996=CARTESIAN_POINT('',(180.0,-21.749999999999996,90.0));
#1997=VERTEX_POINT('',#1996);
#1998=DIRECTION('',(0.0,-1.0,0.0));
#1999=VECTOR('',#1998,18.250000000000011);
#2000=LINE('',#1996,#1999);
#2001=EDGE_CURVE('',#1997,#1666,#2000,.T.);
#2002=ORIENTED_EDGE('',*,*,#2001,.F.);
#2003=CARTESIAN_POINT('',(140.0,-21.750000000000000,90.0));
#2004=VERTEX_POINT('',#2003);
#2005=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2006=DIRECTION('',(0.0,-1.0,0.0));
#2007=DIRECTION('',(1.0,0.0,0.0));
#2008=AXIS2_PLACEMENT_3D('',#2005,#2006,#2007);
#2009=CIRCLE('',#2008,20.0);
#2010=EDGE_CURVE('',#1997,#2004,#2009,.T.);
#2011=ORIENTED_EDGE('',*,*,#2010,.T.);
#2012=DIRECTION('',(0.0,-1.0,0.0));
#2013=VECTOR('',#2012,18.250000000000007);
#2014=LINE('',#2003,#2013);
#2015=EDGE_CURVE('',#2004,#1668,#2014,.T.);
#2016=ORIENTED_EDGE('',*,*,#2015,.T.);
#2017=EDGE_LOOP('',(#1995,#2002,#2011,#2016));
#2018=FACE_BOUND('',#2017,.F.);
#2019=ADVANCED_FACE('',(#2018),#1989,.F.);
#2020=SHAPE_ASPECT('featured shape: face 23',$,#38,.T.);
#2021=PROPERTY_DEFINITION('',$,#2020);
#2022=FACE_SHAPE_REPRESENTATION('',(#2019),#29);
#2023=PROPERTY_DEFINITION_REPRESENTATION(#2021,#2022);
#2024=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2025=DIRECTION('',(0.0,-1.0,0.0));
#2026=DIRECTION('',(-1.0,0.0,0.0));
#2027=AXIS2_PLACEMENT_3D('',#2024,#2025,#2026);
#2028=CIRCLE('',#2027,20.0);
#2029=EDGE_CURVE('',#1964,#1962,#2028,.T.);

```

```

#2030=ORIENTED_EDGE('',*,*,#2029,.F.);
#2031=ORIENTED_EDGE('',*,*,#1981,.F.);
#2032=ORIENTED_EDGE('',*,*,#800,.T.);
#2033=ORIENTED_EDGE('',*,*,#1975,.T.);
#2034=EDGE_LOOP('',(#2030,#2031,#2032,#2033));
#2035=FACE_BOUND('',#2034,.F.);
#2036=CARTESIAN_POINT('',(160.0,0.0,90.0));
#2037=DIRECTION('',(0.0,-1.0,0.0));
#2038=DIRECTION('',(1.0,0.0,0.0));
#2039=AXIS2_PLACEMENT_3D('',#2036,#2037,#2038);
#2040=CYLINDRICAL_SURFACE('',#2039,20.0);
#2041=ADVANCED_FACE('',(#2035),#2040,.F.);
#2042=SHAPE_ASPECT('featured shape: face 24',$,#38,.T.);
#2043=PROPERTY_DEFINITION('',$,#2042);
#2044=FACE_SHAPE_REPRESENTATION('',(#2041),#29);
#2045=PROPERTY_DEFINITION_REPRESENTATION(#2043,#2044);
#2046=ORIENTED_EDGE('',*,*,#1681,.F.);
#2047=ORIENTED_EDGE('',*,*,#2015,.F.);
#2048=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2049=DIRECTION('',(0.0,-1.0,0.0));
#2050=DIRECTION('',(-1.0,0.0,0.0));
#2051=AXIS2_PLACEMENT_3D('',#2048,#2049,#2050);
#2052=CIRCLE('',#2051,20.0);
#2053=EDGE_CURVE('',#2004,#1997,#2052,.T.);
#2054=ORIENTED_EDGE('',*,*,#2053,.T.);
#2055=ORIENTED_EDGE('',*,*,#2001,.T.);
#2056=EDGE_LOOP('',(#2046,#2047,#2054,#2055));
#2057=FACE_BOUND('',#2056,.F.);
#2058=ADVANCED_FACE('',(#2057),#2040,.F.);
#2059=SHAPE_ASPECT('featured shape: face 25',$,#38,.T.);
#2060=PROPERTY_DEFINITION('',$,#2059);
#2061=FACE_SHAPE_REPRESENTATION('',(#2058),#29);
#2062=PROPERTY_DEFINITION_REPRESENTATION(#2060,#2061);
#2063=CARTESIAN_POINT('',(50.0,-10.0,99.0));
#2064=VERTEX_POINT('',#2063);
#2065=DIRECTION('',(1.0,0.0,0.0));
#2066=VECTOR('',#2065,50.0);
#2067=LINE('',#156,#2066);
#2068=EDGE_CURVE('',#157,#2064,#2067,.T.);
#2069=ORIENTED_EDGE('',*,*,#2068,.T.);
#2070=DIRECTION('',(0.0,-1.0,0.0));
#2071=VECTOR('',#2070,10.0);
#2072=LINE('',#698,#2071);
#2073=EDGE_CURVE('',#699,#2064,#2072,.T.);
#2074=ORIENTED_EDGE('',*,*,#2073,.F.);
#2075=ORIENTED_EDGE('',*,*,#703,.F.);
#2076=ORIENTED_EDGE('',*,*,#168,.T.);
#2077=EDGE_LOOP('',(#2069,#2074,#2075,#2076));
#2078=FACE_BOUND('',#2077,.F.);
#2079=CARTESIAN_POINT('',(0.0,0.0,99.0));
#2080=DIRECTION('',(0.0,0.0,1.0));
#2081=DIRECTION('',(1.0,0.0,0.0));
#2082=AXIS2_PLACEMENT_3D('',#2079,#2080,#2081);

```

ISO 10303-223:2008 (E)

```

#2083=PLANE('', #2082);
#2084=ADVANCED_FACE('', (#2078), #2083, .F.);
#2085=SHAPE_ASPECT('featured shape: face 26', $, #38, .T.);
#2086=PROPERTY_DEFINITION('', $, #2085);
#2087=FACE_SHAPE_REPRESENTATION('', (#2084), #29);
#2088=PROPERTY_DEFINITION_REPRESENTATION(#2086, #2087);
#2089=CARTESIAN_POINT('', (50.0, -10.0, 81.0));
#2090=VERTEX_POINT('', #2089);
#2091=DIRECTION('', (-1.0, 0.0, 0.0));
#2092=VECTOR('', #2091, 50.0);
#2093=LINE('', #2089, #2092);
#2094=EDGE_CURVE('', #2090, #206, #2093, .T.);
#2095=ORIENTED_EDGE('', *, *, #2094, .T.);
#2096=ORIENTED_EDGE('', *, *, #210, .F.);
#2097=ORIENTED_EDGE('', *, *, #717, .F.);
#2098=DIRECTION('', (0.0, -1.0, 0.0));
#2099=VECTOR('', #2098, 10.0);
#2100=LINE('', #705, #2099);
#2101=EDGE_CURVE('', #706, #2090, #2100, .T.);
#2102=ORIENTED_EDGE('', *, *, #2101, .T.);
#2103=EDGE_LOOP('', (#2095, #2096, #2097, #2102));
#2104=FACE_BOUND('', #2103, .F.);
#2105=CARTESIAN_POINT('', (50.0, 0.0, 81.0));
#2106=DIRECTION('', (0.0, 0.0, -1.0));
#2107=DIRECTION('', (-1.0, 0.0, 0.0));
#2108=AXIS2_PLACEMENT_3D('', #2105, #2106, #2107);
#2109=PLANE('', #2108);
#2110=ADVANCED_FACE('', (#2104), #2109, .F.);
#2111=SHAPE_ASPECT('featured shape: face 27', $, #38, .T.);
#2112=PROPERTY_DEFINITION('', $, #2111);
#2113=FACE_SHAPE_REPRESENTATION('', (#2110), #29);
#2114=PROPERTY_DEFINITION_REPRESENTATION(#2112, #2113);
#2115=DIRECTION('', (0.0, 1.0, 0.0));
#2116=VECTOR('', #2115, 10.0);
#2117=LINE('', #364, #2116);
#2118=EDGE_CURVE('', #365, #1514, #2117, .T.);
#2119=ORIENTED_EDGE('', *, *, #2118, .T.);
#2120=ORIENTED_EDGE('', *, *, #1520, .T.);
#2121=DIRECTION('', (0.0, 1.0, 0.0));
#2122=VECTOR('', #2121, 10.0);
#2123=LINE('', #373, #2122);
#2124=EDGE_CURVE('', #374, #1516, #2123, .T.);
#2125=ORIENTED_EDGE('', *, *, #2124, .F.);
#2126=ORIENTED_EDGE('', *, *, #378, .T.);
#2127=EDGE_LOOP('', (#2119, #2120, #2125, #2126));
#2128=FACE_BOUND('', #2127, .F.);
#2129=CARTESIAN_POINT('', (228.0, -50.0, 180.0));
#2130=DIRECTION('', (-1.0, 0.0, 0.0));
#2131=DIRECTION('', (0.0, 1.0, 0.0));
#2132=AXIS2_PLACEMENT_3D('', #2129, #2130, #2131);
#2133=PLANE('', #2132);
#2134=ADVANCED_FACE('', (#2128), #2133, .F.);
#2135=SHAPE_ASPECT('featured shape: face 28', $, #38, .T.);

```



```

#2136=PROPERTY_DEFINITION('', $, #2135);
#2137=FACE_SHAPE_REPRESENTATION('', (#2134), #29);
#2138=PROPERTY_DEFINITION_REPRESENTATION(#2136, #2137);
#2139=DIRECTION('', (0.0, 1.0, 0.0));
#2140=VECTOR('', #2139, 10.0);
#2141=LINE('', #480, #2140);
#2142=EDGE_CURVE('', #481, #1621, #2141, .T.);
#2143=ORIENTED_EDGE('', *, *, #2142, .F.);
#2144=ORIENTED_EDGE('', *, *, #485, .T.);
#2145=DIRECTION('', (0.0, 1.0, 0.0));
#2146=VECTOR('', #2145, 10.0);
#2147=LINE('', #471, #2146);
#2148=EDGE_CURVE('', #472, #1630, #2147, .T.);
#2149=ORIENTED_EDGE('', *, *, #2148, .T.);
#2150=ORIENTED_EDGE('', *, *, #1634, .T.);
#2151=EDGE_LOOP('', (#2143, #2144, #2149, #2150));
#2152=FACE_BOUND('', #2151, .F.);
#2153=ADVANCED_FACE('', (#2152), #2133, .F.);
#2154=SHAPE_ASPECT('featured shape: face 29', $, #38, .T.);
#2155=PROPERTY_DEFINITION('', $, #2154);
#2156=FACE_SHAPE_REPRESENTATION('', (#2153), #29);
#2157=PROPERTY_DEFINITION_REPRESENTATION(#2155, #2156);
#2158=DIRECTION('', (0.0, 1.0, 0.0));
#2159=VECTOR('', #2158, 10.0);
#2160=LINE('', #293, #2159);
#2161=EDGE_CURVE('', #294, #1489, #2160, .T.);
#2162=ORIENTED_EDGE('', *, *, #2161, .F.);
#2163=ORIENTED_EDGE('', *, *, #298, .F.);
#2164=DIRECTION('', (0.0, 1.0, 0.0));
#2165=VECTOR('', #2164, 10.0);
#2166=LINE('', #284, #2165);
#2167=EDGE_CURVE('', #285, #1498, #2166, .T.);
#2168=ORIENTED_EDGE('', *, *, #2167, .T.);
#2169=ORIENTED_EDGE('', *, *, #1502, .T.);
#2170=EDGE_LOOP('', (#2162, #2163, #2168, #2169));
#2171=FACE_BOUND('', #2170, .F.);
#2172=CARTESIAN_POINT('', (2.0, -50.0, 0.0));
#2173=DIRECTION('', (1.0, 0.0, 0.0));
#2174=DIRECTION('', (0.0, 1.0, 0.0));
#2175=AXIS2_PLACEMENT_3D('', #2172, #2173, #2174);
#2176=PLANE('', #2175);
#2177=ADVANCED_FACE('', (#2171), #2176, .F.);
#2178=SHAPE_ASPECT('featured shape: face 30', $, #38, .T.);
#2179=PROPERTY_DEFINITION('', $, #2178);
#2180=FACE_SHAPE_REPRESENTATION('', (#2177), #29);
#2181=PROPERTY_DEFINITION_REPRESENTATION(#2179, #2180);
#2182=DIRECTION('', (0.0, 1.0, 0.0));
#2183=VECTOR('', #2182, 10.0);
#2184=LINE('', #546, #2183);
#2185=EDGE_CURVE('', #547, #1726, #2184, .T.);
#2186=ORIENTED_EDGE('', *, *, #2185, .T.);
#2187=ORIENTED_EDGE('', *, *, #1732, .T.);
#2188=DIRECTION('', (0.0, 1.0, 0.0));

```

ISO 10303-223:2008 (E)

```

#2189=VECTOR('', #2188, 10.0);
#2190=LINE('', #555, #2189);
#2191=EDGE_CURVE('', #556, #1728, #2190, .T.);
#2192=ORIENTED_EDGE('', *, *, #2191, .F.);
#2193=ORIENTED_EDGE('', *, *, #560, .F.);
#2194=EDGE_LOOP('', (#2186, #2187, #2192, #2193));
#2195=FACE_BOUND('', #2194, .F.);
#2196=ADVANCED_FACE('', (#2195), #2176, .F.);
#2197=SHAPE_ASPECT('featured shape: face 31', $, #38, .T.);
#2198=PROPERTY_DEFINITION('', $, #2197);
#2199=FACE_SHAPE_REPRESENTATION('', (#2196), #29);
#2200=PROPERTY_DEFINITION_REPRESENTATION(#2198, #2199);
#2201=DIRECTION('', (0.0, 1.0, 0.0));
#2202=VECTOR('', #2201, 10.0);
#2203=LINE('', #455, #2202);
#2204=EDGE_CURVE('', #456, #1580, #2203, .T.);
#2205=ORIENTED_EDGE('', *, *, #2204, .T.);
#2206=ORIENTED_EDGE('', *, *, #1586, .T.);
#2207=DIRECTION('', (0.0, 1.0, 0.0));
#2208=VECTOR('', #2207, 10.0);
#2209=LINE('', #464, #2208);
#2210=EDGE_CURVE('', #465, #1582, #2209, .T.);
#2211=ORIENTED_EDGE('', *, *, #2210, .F.);
#2212=ORIENTED_EDGE('', *, *, #469, .F.);
#2213=EDGE_LOOP('', (#2205, #2206, #2211, #2212));
#2214=FACE_BOUND('', #2213, .F.);
#2215=CARTESIAN_POINT('', (0.0, -50.0, 178.0));
#2216=DIRECTION('', (0.0, 0.0, -1.0));
#2217=DIRECTION('', (0.0, 1.0, 0.0));
#2218=AXIS2_PLACEMENT_3D('', #2215, #2216, #2217);
#2219=PLANE('', #2218);
#2220=ADVANCED_FACE('', (#2214), #2219, .F.);
#2221=SHAPE_ASPECT('featured shape: face 32', $, #38, .T.);
#2222=PROPERTY_DEFINITION('', $, #2221);
#2223=FACE_SHAPE_REPRESENTATION('', (#2220), #29);
#2224=PROPERTY_DEFINITION_REPRESENTATION(#2222, #2223);
#2225=DIRECTION('', (0.0, 1.0, 0.0));
#2226=VECTOR('', #2225, 10.0);
#2227=LINE('', #571, #2226);
#2228=EDGE_CURVE('', #572, #1767, #2227, .T.);
#2229=ORIENTED_EDGE('', *, *, #2228, .F.);
#2230=ORIENTED_EDGE('', *, *, #576, .F.);
#2231=DIRECTION('', (0.0, 1.0, 0.0));
#2232=VECTOR('', #2231, 10.0);
#2233=LINE('', #562, #2232);
#2234=EDGE_CURVE('', #563, #1776, #2233, .T.);
#2235=ORIENTED_EDGE('', *, *, #2234, .T.);
#2236=ORIENTED_EDGE('', *, *, #1780, .T.);
#2237=EDGE_LOOP('', (#2229, #2230, #2235, #2236));
#2238=FACE_BOUND('', #2237, .F.);
#2239=ADVANCED_FACE('', (#2238), #2219, .F.);
#2240=SHAPE_ASPECT('featured shape: face 33', $, #38, .T.);
#2241=PROPERTY_DEFINITION('', $, #2240);

```

```

#2242=FACE_SHAPE_REPRESENTATION('', (#2239), #29);
#2243=PROPERTY_DEFINITION_REPRESENTATION(#2241, #2242);
#2244=DIRECTION('', (0.0, 1.0, 0.0));
#2245=VECTOR('', #2244, 10.0);
#2246=LINE('', #268, #2245);
#2247=EDGE_CURVE('', #269, #1448, #2246, .T.);
#2248=ORIENTED_EDGE('', *, *, #2247, .T.);
#2249=ORIENTED_EDGE('', *, *, #1454, .T.);
#2250=DIRECTION('', (0.0, 1.0, 0.0));
#2251=VECTOR('', #2250, 10.0);
#2252=LINE('', #277, #2251);
#2253=EDGE_CURVE('', #278, #1450, #2252, .T.);
#2254=ORIENTED_EDGE('', *, *, #2253, .F.);
#2255=ORIENTED_EDGE('', *, *, #282, .T.);
#2256=EDGE_LOOP('', (#2248, #2249, #2254, #2255));
#2257=FACE_BOUND('', #2256, .F.);
#2258=CARTESIAN_POINT('', (230.0, -50.0, 2.0));
#2259=DIRECTION('', (0.0, 0.0, 1.0));
#2260=DIRECTION('', (0.0, 1.0, 0.0));
#2261=AXIS2_PLACEMENT_3D('', #2258, #2259, #2260);
#2262=PLANE('', #2261);
#2263=ADVANCED_FACE('', (#2257), #2262, .F.);
#2264=SHAPE_ASPECT('featured shape: face 34', $, #38, .T.);
#2265=PROPERTY_DEFINITION('', $, #2264);
#2266=FACE_SHAPE_REPRESENTATION('', (#2263), #29);
#2267=PROPERTY_DEFINITION_REPRESENTATION(#2265, #2266);
#2268=DIRECTION('', (0.0, 1.0, 0.0));
#2269=VECTOR('', #2268, 10.0);
#2270=LINE('', #389, #2269);
#2271=EDGE_CURVE('', #390, #1555, #2270, .T.);
#2272=ORIENTED_EDGE('', *, *, #2271, .F.);
#2273=ORIENTED_EDGE('', *, *, #394, .T.);
#2274=DIRECTION('', (0.0, 1.0, 0.0));
#2275=VECTOR('', #2274, 10.0);
#2276=LINE('', #380, #2275);
#2277=EDGE_CURVE('', #381, #1564, #2276, .T.);
#2278=ORIENTED_EDGE('', *, *, #2277, .T.);
#2279=ORIENTED_EDGE('', *, *, #1568, .T.);
#2280=EDGE_LOOP('', (#2272, #2273, #2278, #2279));
#2281=FACE_BOUND('', #2280, .F.);
#2282=ADVANCED_FACE('', (#2281), #2262, .F.);
#2283=SHAPE_ASPECT('featured shape: face 35', $, #38, .T.);
#2284=PROPERTY_DEFINITION('', $, #2283);
#2285=FACE_SHAPE_REPRESENTATION('', (#2282), #29);
#2286=PROPERTY_DEFINITION_REPRESENTATION(#2284, #2285);
#2287=CARTESIAN_POINT('', (182.0, -18.250000000000000, 90.0));
#2288=VERTEX_POINT('', #2287);
#2289=CARTESIAN_POINT('', (138.0, -18.250000000000000, 90.0));
#2290=VERTEX_POINT('', #2289);
#2291=CARTESIAN_POINT('', (160.0, -18.250000000000000, 90.0));
#2292=DIRECTION('', (0.0, 1.0, 0.0));
#2293=DIRECTION('', (1.0, 0.0, 0.0));
#2294=AXIS2_PLACEMENT_3D('', #2291, #2292, #2293);

```

```

#2295=CIRCLE('',#2294,22.0);
#2296=EDGE_CURVE('',#2288,#2290,#2295,.T.);
#2297=ORIENTED_EDGE('',*,*,#2296,.T.);
#2298=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2299=DIRECTION('',(0.0,-1.0,0.0));
#2300=DIRECTION('',(1.0,0.0,0.0));
#2301=AXIS2_PLACEMENT_3D('',#2298,#2299,#2300);
#2302=CIRCLE('',#2301,22.0);
#2303=EDGE_CURVE('',#2288,#2290,#2302,.T.);
#2304=ORIENTED_EDGE('',*,*,#2303,.F.);
#2305=EDGE_LOOP('',(#2297,#2304));
#2306=FACE_BOUND('',#2305,.F.);
#2307=ORIENTED_EDGE('',*,*,#1970,.T.);
#2308=ORIENTED_EDGE('',*,*,#2029,.T.);
#2309=EDGE_LOOP('',(#2307,#2308));
#2310=FACE_BOUND('',#2309,.F.);
#2311=CARTESIAN_POINT('',(160.0,-18.250000000000000,90.0));
#2312=DIRECTION('',(0.0,-1.0,0.0));
#2313=DIRECTION('',(1.0,0.0,0.0));
#2314=AXIS2_PLACEMENT_3D('',#2311,#2312,#2313);
#2315=PLANE('',#2314);
#2316=ADVANCED_FACE('',(#2306,#2310),#2315,.T.);
#2317=SHAPE_ASPECT('featured shape: face 36',$,#38,.T.);
#2318=PROPERTY_DEFINITION('',$,#2317);
#2319=FACE_SHAPE_REPRESENTATION('',(#2316),#29);
#2320=PROPERTY_DEFINITION_REPRESENTATION(#2318,#2319);
#2324=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MICRO.,.METRE.));
#2329=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(1.600000000000000),#2324)REPRESENTATION_ITEM('surface
finish'));
#2330=REPRESENTATION('surface finish value',(#2329),#29);
#2331=PROPERTY_DEFINITION('surface property','surface finish',#2317);
#2332=PROPERTY_DEFINITION('surface property','surface finish',#2317);
#2333=PROPERTY_DEFINITION_REPRESENTATION(#2332,#2330);
#2334=PROPERTY_DEFINITION_RELATIONSHIP('',',',#2331,#2332);
#2335=CARTESIAN_POINT('',(182.0,-21.750000000000000,90.0));
#2336=VERTEX_POINT('',#2335);
#2337=DIRECTION('',(0.0,-1.0,0.0));
#2338=VECTOR('',#2337,3.500000000000000);
#2339=LINE('',#2287,#2338);
#2340=EDGE_CURVE('',#2288,#2336,#2339,.T.);
#2341=ORIENTED_EDGE('',*,*,#2340,.F.);
#2342=ORIENTED_EDGE('',*,*,#2303,.T.);
#2343=CARTESIAN_POINT('',(138.0,-21.750000000000000,90.0));
#2344=VERTEX_POINT('',#2343);
#2345=DIRECTION('',(0.0,-1.0,0.0));
#2346=VECTOR('',#2345,3.500000000000000);
#2347=LINE('',#2289,#2346);
#2348=EDGE_CURVE('',#2290,#2344,#2347,.T.);
#2349=ORIENTED_EDGE('',*,*,#2348,.T.);
#2350=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2351=DIRECTION('',(0.0,-1.0,0.0));
#2352=DIRECTION('',(1.0,0.0,0.0));

```

```

#2353=AXIS2_PLACEMENT_3D('',#2350,#2351,#2352);
#2354=CIRCLE('',#2353,22.0);
#2355=EDGE_CURVE('',#2336,#2344,#2354,.T.);
#2356=ORIENTED_EDGE('',*,*,#2355,.F.);
#2357=EDGE_LOOP('',(#2341,#2342,#2349,#2356));
#2358=FACE_BOUND('',#2357,.F.);
#2359=CARTESIAN_POINT('',(160.0,-18.074999999999999,90.0));
#2360=DIRECTION('',(0.0,-1.0,0.0));
#2361=DIRECTION('',(1.0,0.0,0.0));
#2362=AXIS2_PLACEMENT_3D('',#2359,#2360,#2361);
#2363=CYLINDRICAL_SURFACE('',#2362,22.0);
#2364=ADVANCED_FACE('',(#2358),#2363,.F.);
#2365=SHAPE_ASPECT('featured shape: face 37',$,#38,.T.);
#2366=PROPERTY_DEFINITION('',$,#2365);
#2367=FACE_SHAPE_REPRESENTATION('',(#2364),#29);
#2368=PROPERTY_DEFINITION_REPRESENTATION(#2366,#2367);
#2369=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2370=DIRECTION('',(0.0,1.0,0.0));
#2371=DIRECTION('',(1.0,0.0,0.0));
#2372=AXIS2_PLACEMENT_3D('',#2369,#2370,#2371);
#2373=CIRCLE('',#2372,22.0);
#2374=EDGE_CURVE('',#2336,#2344,#2373,.T.);
#2375=ORIENTED_EDGE('',*,*,#2374,.F.);
#2376=ORIENTED_EDGE('',*,*,#2355,.T.);
#2377=EDGE_LOOP('',(#2375,#2376));
#2378=FACE_BOUND('',#2377,.F.);
#2379=ORIENTED_EDGE('',*,*,#2010,.F.);
#2380=ORIENTED_EDGE('',*,*,#2053,.F.);
#2381=EDGE_LOOP('',(#2379,#2380));
#2382=FACE_BOUND('',#2381,.F.);
#2383=CARTESIAN_POINT('',(160.0,-21.750000000000000,90.0));
#2384=DIRECTION('',(0.0,-1.0,0.0));
#2385=DIRECTION('',(1.0,0.0,0.0));
#2386=AXIS2_PLACEMENT_3D('',#2383,#2384,#2385);
#2387=PLANE('',#2386);
#2388=ADVANCED_FACE('',(#2378,#2382),#2387,.F.);
#2389=SHAPE_ASPECT('featured shape: face 38',$,#38,.T.);
#2390=PROPERTY_DEFINITION('',$,#2389);
#2391=FACE_SHAPE_REPRESENTATION('',(#2388),#29);
#2392=PROPERTY_DEFINITION_REPRESENTATION(#2390,#2391);
#2393=ORIENTED_EDGE('',*,*,#2340,.T.);
#2394=ORIENTED_EDGE('',*,*,#2374,.T.);
#2395=ORIENTED_EDGE('',*,*,#2348,.F.);
#2396=ORIENTED_EDGE('',*,*,#2296,.F.);
#2397=EDGE_LOOP('',(#2393,#2394,#2395,#2396));
#2398=FACE_BOUND('',#2397,.F.);
#2399=CARTESIAN_POINT('',(160.0,-18.074999999999999,90.0));
#2400=DIRECTION('',(0.0,-1.0,0.0));
#2401=DIRECTION('',(1.0,0.0,0.0));
#2402=AXIS2_PLACEMENT_3D('',#2399,#2400,#2401);
#2403=CYLINDRICAL_SURFACE('',#2402,22.0);
#2404=ADVANCED_FACE('',(#2398),#2403,.F.);
#2405=SHAPE_ASPECT('featured shape: face 39',$,#38,.T.);

```

```

#2406=PROPERTY_DEFINITION('', $, #2405);
#2407=FACE_SHAPE_REPRESENTATION('', (#2404), #29);
#2408=PROPERTY_DEFINITION_REPRESENTATION(#2406, #2407);
#2409=CARTESIAN_POINT('', (167.50000000000000, -30.0, 30.0));
#2410=VERTEX_POINT('', #2409);
#2411=CARTESIAN_POINT('', (152.50000000000000, -30.0, 30.0));
#2412=VERTEX_POINT('', #2411);
#2413=CARTESIAN_POINT('', (160.0, -30.0, 30.0));
#2414=DIRECTION('', (0.0, -1.0, 0.0));
#2415=DIRECTION('', (1.0, 0.0, 0.0));
#2416=AXIS2_PLACEMENT_3D('', #2413, #2414, #2415);
#2417=CIRCLE('', #2416, 7.500000000000000);
#2418=EDGE_CURVE('', #2410, #2412, #2417, .T.);
#2419=ORIENTED_EDGE('', *, *, #2418, .T.);
#2420=CARTESIAN_POINT('', (160.0, -30.0, 30.0));
#2421=DIRECTION('', (0.0, 1.0, 0.0));
#2422=DIRECTION('', (1.0, 0.0, 0.0));
#2423=AXIS2_PLACEMENT_3D('', #2420, #2421, #2422);
#2424=CIRCLE('', #2423, 7.500000000000000);
#2425=EDGE_CURVE('', #2410, #2412, #2424, .T.);
#2426=ORIENTED_EDGE('', *, *, #2425, .F.);
#2427=EDGE_LOOP('', (#2419, #2426));
#2428=FACE_BOUND('', #2427, .F.);
#2429=CARTESIAN_POINT('', (160.0, -30.0, 30.0));
#2430=DIRECTION('', (0.0, 1.0, 0.0));
#2431=DIRECTION('', (1.0, 0.0, 0.0));
#2432=AXIS2_PLACEMENT_3D('', #2429, #2430, #2431);
#2433=PLANE('', #2432);
#2434=ADVANCED_FACE('', (#2428), #2433, .T.);
#2435=SHAPE_ASPECT('featured shape: face 40', $, #38, .T.);
#2436=PROPERTY_DEFINITION('', $, #2435);
#2437=FACE_SHAPE_REPRESENTATION('', (#2434), #29);
#2438=PROPERTY_DEFINITION_REPRESENTATION(#2436, #2437);
#2439=CARTESIAN_POINT('', (167.50000000000000, -15.0, 30.0));
#2440=VERTEX_POINT('', #2439);
#2441=DIRECTION('', (0.0, 1.0, 0.0));
#2442=VECTOR('', #2441, 15.0);
#2443=LINE('', #2409, #2442);
#2444=EDGE_CURVE('', #2410, #2440, #2443, .T.);
#2445=ORIENTED_EDGE('', *, *, #2444, .F.);
#2446=ORIENTED_EDGE('', *, *, #2425, .T.);
#2447=CARTESIAN_POINT('', (152.50000000000000, -15.0, 30.0));
#2448=VERTEX_POINT('', #2447);
#2449=DIRECTION('', (0.0, 1.0, 0.0));
#2450=VECTOR('', #2449, 15.0);
#2451=LINE('', #2411, #2450);
#2452=EDGE_CURVE('', #2412, #2448, #2451, .T.);
#2453=ORIENTED_EDGE('', *, *, #2452, .T.);
#2454=CARTESIAN_POINT('', (160.0, -15.0, 30.0));
#2455=DIRECTION('', (0.0, 1.0, 0.0));
#2456=DIRECTION('', (1.0, 0.0, 0.0));
#2457=AXIS2_PLACEMENT_3D('', #2454, #2455, #2456);
#2458=CIRCLE('', #2457, 7.500000000000000);

```

```

#2459=EDGE_CURVE('',#2440,#2448,#2458,.T.);
#2460=ORIENTED_EDGE('',*,*,#2459,.F.);
#2461=EDGE_LOOP('',(#2445,#2446,#2453,#2460));
#2462=FACE_BOUND('',#2461,.F.);
#2463=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2464=DIRECTION('',(0.0,1.0,0.0));
#2465=DIRECTION('',(1.0,0.0,0.0));
#2466=AXIS2_PLACEMENT_3D('',#2463,#2464,#2465);
#2467=CYLINDRICAL_SURFACE('',#2466,7.500000000000000);
#2468=ADVANCED_FACE('',(#2462),#2467,.F.);
#2469=SHAPE_ASPECT('featured shape: face 41',$,#38,.T.);
#2470=PROPERTY_DEFINITION('',$,#2469);
#2471=FACE_SHAPE_REPRESENTATION('',(#2468),#29);
#2472=PROPERTY_DEFINITION_REPRESENTATION(#2470,#2471);
#2473=CARTESIAN_POINT('',(172.500000000000000,-15.0,30.0));
#2474=VERTEX_POINT('',#2473);
#2475=CARTESIAN_POINT('',(147.500000000000000,-15.0,30.0));
#2476=VERTEX_POINT('',#2475);
#2477=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2478=DIRECTION('',(0.0,1.0,0.0));
#2479=DIRECTION('',(1.0,0.0,0.0));
#2480=AXIS2_PLACEMENT_3D('',#2477,#2478,#2479);
#2481=CIRCLE('',#2480,12.500000000000000);
#2482=EDGE_CURVE('',#2474,#2476,#2481,.T.);
#2483=ORIENTED_EDGE('',*,*,#2482,.F.);
#2484=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2485=DIRECTION('',(0.0,-1.0,0.0));
#2486=DIRECTION('',(1.0,0.0,0.0));
#2487=AXIS2_PLACEMENT_3D('',#2484,#2485,#2486);
#2488=CIRCLE('',#2487,12.500000000000000);
#2489=EDGE_CURVE('',#2474,#2476,#2488,.T.);
#2490=ORIENTED_EDGE('',*,*,#2489,.T.);
#2491=EDGE_LOOP('',(#2483,#2490));
#2492=FACE_BOUND('',#2491,.F.);
#2493=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2494=DIRECTION('',(0.0,-1.0,0.0));
#2495=DIRECTION('',(1.0,0.0,0.0));
#2496=AXIS2_PLACEMENT_3D('',#2493,#2494,#2495);
#2497=CIRCLE('',#2496,7.500000000000000);
#2498=EDGE_CURVE('',#2440,#2448,#2497,.T.);
#2499=ORIENTED_EDGE('',*,*,#2498,.F.);
#2500=ORIENTED_EDGE('',*,*,#2459,.T.);
#2501=EDGE_LOOP('',(#2499,#2500));
#2502=FACE_BOUND('',#2501,.F.);
#2503=CARTESIAN_POINT('',(160.0,-15.0,30.0));
#2504=DIRECTION('',(0.0,1.0,0.0));
#2505=DIRECTION('',(1.0,0.0,0.0));
#2506=AXIS2_PLACEMENT_3D('',#2503,#2504,#2505);
#2507=PLANE('',#2506);
#2508=ADVANCED_FACE('',(#2492,#2502),#2507,.T.);
#2509=SHAPE_ASPECT('featured shape: face 42',$,#38,.T.);
#2510=PROPERTY_DEFINITION('',$,#2509);
#2511=FACE_SHAPE_REPRESENTATION('',(#2508),#29);

```

```

#2512=PROPERTY_DEFINITION_REPRESENTATION(#2510,#2511);
#2513=ORIENTED_EDGE('',*,*,#813,.F.);
#2514=DIRECTION('',(0.0,1.0,0.0));
#2515=VECTOR('',#2514,15.0);
#2516=LINE('',#2473,#2515);
#2517=EDGE_CURVE('',#2474,#805,#2516,.T.);
#2518=ORIENTED_EDGE('',*,*,#2517,.F.);
#2519=ORIENTED_EDGE('',*,*,#2482,.T.);
#2520=DIRECTION('',(0.0,1.0,0.0));
#2521=VECTOR('',#2520,15.0);
#2522=LINE('',#2475,#2521);
#2523=EDGE_CURVE('',#2476,#807,#2522,.T.);
#2524=ORIENTED_EDGE('',*,*,#2523,.T.);
#2525=EDGE_LOOP('',(#2513,#2518,#2519,#2524));
#2526=FACE_BOUND('',#2525,.F.);
#2527=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2528=DIRECTION('',(0.0,1.0,0.0));
#2529=DIRECTION('',(1.0,0.0,0.0));
#2530=AXIS2_PLACEMENT_3D('',#2527,#2528,#2529);
#2531=CYLINDRICAL_SURFACE('',#2530,12.500000000000000);
#2532=ADVANCED_FACE('',(#2526),#2531,.F.);
#2533=SHAPE_ASPECT('featured shape: face 43',$,#38,.T.);
#2534=PROPERTY_DEFINITION('',$,#2533);
#2535=FACE_SHAPE_REPRESENTATION('',(#2532),#29);
#2536=PROPERTY_DEFINITION_REPRESENTATION(#2534,#2535);
#2537=ORIENTED_EDGE('',*,*,#2444,.T.);
#2538=ORIENTED_EDGE('',*,*,#2498,.T.);
#2539=ORIENTED_EDGE('',*,*,#2452,.F.);
#2540=ORIENTED_EDGE('',*,*,#2418,.F.);
#2541=EDGE_LOOP('',(#2537,#2538,#2539,#2540));
#2542=FACE_BOUND('',#2541,.F.);
#2543=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2544=DIRECTION('',(0.0,1.0,0.0));
#2545=DIRECTION('',(1.0,0.0,0.0));
#2546=AXIS2_PLACEMENT_3D('',#2543,#2544,#2545);
#2547=CYLINDRICAL_SURFACE('',#2546,7.500000000000000);
#2548=ADVANCED_FACE('',(#2542),#2547,.F.);
#2549=SHAPE_ASPECT('featured shape: face 44',$,#38,.T.);
#2550=PROPERTY_DEFINITION('',$,#2549);
#2551=FACE_SHAPE_REPRESENTATION('',(#2548),#29);
#2552=PROPERTY_DEFINITION_REPRESENTATION(#2550,#2551);
#2553=ORIENTED_EDGE('',*,*,#820,.F.);
#2554=ORIENTED_EDGE('',*,*,#2523,.F.);
#2555=ORIENTED_EDGE('',*,*,#2489,.F.);
#2556=ORIENTED_EDGE('',*,*,#2517,.T.);
#2557=EDGE_LOOP('',(#2553,#2554,#2555,#2556));
#2558=FACE_BOUND('',#2557,.F.);
#2559=CARTESIAN_POINT('',(160.0,-31.500000000000000,30.0));
#2560=DIRECTION('',(0.0,1.0,0.0));
#2561=DIRECTION('',(1.0,0.0,0.0));
#2562=AXIS2_PLACEMENT_3D('',#2559,#2560,#2561);
#2563=CYLINDRICAL_SURFACE('',#2562,12.500000000000000);
#2564=ADVANCED_FACE('',(#2558),#2563,.F.);

```



```

#2565=SHAPE_ASPECT('featured shape: face 45',$,#38,.T.);
#2566=PROPERTY_DEFINITION('', $, #2565);
#2567=FACE_SHAPE_REPRESENTATION('', (#2564), #29);
#2568=PROPERTY_DEFINITION_REPRESENTATION(#2566, #2567);
#2569=ORIENTED_EDGE('', *, *, #1654, .T.);
#2570=CARTESIAN_POINT('', (152.50000000000000, -15.0, 150.0));
#2571=VERTEX_POINT('', #2570);
#2572=DIRECTION('', (0.0, 1.0, 0.0));
#2573=VECTOR('', #2572, 25.0);
#2574=LINE('', #1647, #2573);
#2575=EDGE_CURVE('', #1648, #2571, #2574, .T.);
#2576=ORIENTED_EDGE('', *, *, #2575, .T.);
#2577=CARTESIAN_POINT('', (167.50000000000000, -15.0, 150.0));
#2578=VERTEX_POINT('', #2577);
#2579=CARTESIAN_POINT('', (160.0, -15.0, 150.0));
#2580=DIRECTION('', (0.0, 1.0, 0.0));
#2581=DIRECTION('', (1.0, 0.0, 0.0));
#2582=AXIS2_PLACEMENT_3D('', #2579, #2580, #2581);
#2583=CIRCLE('', #2582, 7.500000000000000);
#2584=EDGE_CURVE('', #2578, #2571, #2583, .T.);
#2585=ORIENTED_EDGE('', *, *, #2584, .F.);
#2586=DIRECTION('', (0.0, 1.0, 0.0));
#2587=VECTOR('', #2586, 25.0);
#2588=LINE('', #1645, #2587);
#2589=EDGE_CURVE('', #1646, #2578, #2588, .T.);
#2590=ORIENTED_EDGE('', *, *, #2589, .F.);
#2591=EDGE_LOOP('', (#2569, #2576, #2585, #2590));
#2592=FACE_BOUND('', #2591, .F.);
#2593=CARTESIAN_POINT('', (160.0, -42.0, 150.0));
#2594=DIRECTION('', (0.0, 1.0, 0.0));
#2595=DIRECTION('', (1.0, 0.0, 0.0));
#2596=AXIS2_PLACEMENT_3D('', #2593, #2594, #2595);
#2597=CYLINDRICAL_SURFACE('', #2596, 7.500000000000000);
#2598=ADVANCED_FACE('', (#2592), #2597, .F.);
#2599=SHAPE_ASPECT('featured shape: face 46',$,#38,.T.);
#2600=PROPERTY_DEFINITION('', $, #2599);
#2601=FACE_SHAPE_REPRESENTATION('', (#2598), #29);
#2602=PROPERTY_DEFINITION_REPRESENTATION(#2600, #2601);
#2603=CARTESIAN_POINT('', (172.50000000000000, -15.0, 150.0));
#2604=VERTEX_POINT('', #2603);
#2605=CARTESIAN_POINT('', (147.50000000000000, -15.0, 150.0));
#2606=VERTEX_POINT('', #2605);
#2607=CARTESIAN_POINT('', (160.0, -15.0, 150.0));
#2608=DIRECTION('', (0.0, 1.0, 0.0));
#2609=DIRECTION('', (1.0, 0.0, 0.0));
#2610=AXIS2_PLACEMENT_3D('', #2607, #2608, #2609);
#2611=CIRCLE('', #2610, 12.500000000000000);
#2612=EDGE_CURVE('', #2604, #2606, #2611, .T.);
#2613=ORIENTED_EDGE('', *, *, #2612, .F.);
#2614=CARTESIAN_POINT('', (160.0, -15.0, 150.0));
#2615=DIRECTION('', (0.0, -1.0, 0.0));
#2616=DIRECTION('', (1.0, 0.0, 0.0));
#2617=AXIS2_PLACEMENT_3D('', #2614, #2615, #2616);

```

```

#2618=CIRCLE('',#2617,12.500000000000000);
#2619=EDGE_CURVE('',#2604,#2606,#2618,.T.);
#2620=ORIENTED_EDGE('',*,*,#2619,.T.);
#2621=EDGE_LOOP('',(#2613,#2620));
#2622=FACE_BOUND('',#2621,.F.);
#2623=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2624=DIRECTION('',(0.0,-1.0,0.0));
#2625=DIRECTION('',(1.0,0.0,0.0));
#2626=AXIS2_PLACEMENT_3D('',#2623,#2624,#2625);
#2627=CIRCLE('',#2626,7.500000000000000);
#2628=EDGE_CURVE('',#2578,#2571,#2627,.T.);
#2629=ORIENTED_EDGE('',*,*,#2628,.F.);
#2630=ORIENTED_EDGE('',*,*,#2584,.T.);
#2631=EDGE_LOOP('',(#2629,#2630));
#2632=FACE_BOUND('',#2631,.F.);
#2633=CARTESIAN_POINT('',(160.0,-15.0,150.0));
#2634=DIRECTION('',(0.0,1.0,0.0));
#2635=DIRECTION('',(1.0,0.0,0.0));
#2636=AXIS2_PLACEMENT_3D('',#2633,#2634,#2635);
#2637=PLANE('',#2636);
#2638=ADVANCED_FACE('',(#2622,#2632),#2637,.T.);
#2639=SHAPE_ASPECT('featured shape: face 47',$,#38,.T.);
#2640=PROPERTY_DEFINITION('',$,#2639);
#2641=FACE_SHAPE_REPRESENTATION('',(#2638),#29);
#2642=PROPERTY_DEFINITION_REPRESENTATION(#2640,#2641);
#2643=ORIENTED_EDGE('',*,*,#833,.F.);
#2644=DIRECTION('',(0.0,1.0,0.0));
#2645=VECTOR('',#2644,15.0);
#2646=LINE('',#2603,#2645);
#2647=EDGE_CURVE('',#2604,#825,#2646,.T.);
#2648=ORIENTED_EDGE('',*,*,#2647,.F.);
#2649=ORIENTED_EDGE('',*,*,#2612,.T.);
#2650=DIRECTION('',(0.0,1.0,0.0));
#2651=VECTOR('',#2650,15.0);
#2652=LINE('',#2605,#2651);
#2653=EDGE_CURVE('',#2606,#827,#2652,.T.);
#2654=ORIENTED_EDGE('',*,*,#2653,.T.);
#2655=EDGE_LOOP('',(#2643,#2648,#2649,#2654));
#2656=FACE_BOUND('',#2655,.F.);
#2657=CARTESIAN_POINT('',(160.0,-42.0,150.0));
#2658=DIRECTION('',(0.0,1.0,0.0));
#2659=DIRECTION('',(1.0,0.0,0.0));
#2660=AXIS2_PLACEMENT_3D('',#2657,#2658,#2659);
#2661=CYLINDRICAL_SURFACE('',#2660,12.500000000000000);
#2662=ADVANCED_FACE('',(#2656),#2661,.F.);
#2663=SHAPE_ASPECT('featured shape: face 48',$,#38,.T.);
#2664=PROPERTY_DEFINITION('',$,#2663);
#2665=FACE_SHAPE_REPRESENTATION('',(#2662),#29);
#2666=PROPERTY_DEFINITION_REPRESENTATION(#2664,#2665);
#2667=ORIENTED_EDGE('',*,*,#1661,.T.);
#2668=ORIENTED_EDGE('',*,*,#2589,.T.);
#2669=ORIENTED_EDGE('',*,*,#2628,.T.);
#2670=ORIENTED_EDGE('',*,*,#2575,.F.);

```

```

#2671=EDGE_LOOP('', (#2667, #2668, #2669, #2670));
#2672=FACE_BOUND('', #2671, .F.);
#2673=CARTESIAN_POINT('', (160.0, -42.0, 150.0));
#2674=DIRECTION('', (0.0, 1.0, 0.0));
#2675=DIRECTION('', (1.0, 0.0, 0.0));
#2676=AXIS2_PLACEMENT_3D('', #2673, #2674, #2675);
#2677=CYLINDRICAL_SURFACE('', #2676, 7.500000000000000);
#2678=ADVANCED_FACE('', (#2672), #2677, .F.);
#2679=SHAPE_ASPECT('featured shape: face 49', $, #38, .T.);
#2680=PROPERTY_DEFINITION('', $, #2679);
#2681=FACE_SHAPE_REPRESENTATION('', (#2678), #29);
#2682=PROPERTY_DEFINITION_REPRESENTATION(#2680, #2681);
#2683=ORIENTED_EDGE('', *, *, #840, .F.);
#2684=ORIENTED_EDGE('', *, *, #2653, .F.);
#2685=ORIENTED_EDGE('', *, *, #2619, .F.);
#2686=ORIENTED_EDGE('', *, *, #2647, .T.);
#2687=EDGE_LOOP('', (#2683, #2684, #2685, #2686));
#2688=FACE_BOUND('', #2687, .F.);
#2689=CARTESIAN_POINT('', (160.0, -42.0, 150.0));
#2690=DIRECTION('', (0.0, 1.0, 0.0));
#2691=DIRECTION('', (1.0, 0.0, 0.0));
#2692=AXIS2_PLACEMENT_3D('', #2689, #2690, #2691);
#2693=CYLINDRICAL_SURFACE('', #2692, 12.500000000000000);
#2694=ADVANCED_FACE('', (#2688), #2693, .F.);
#2695=SHAPE_ASPECT('featured shape: face 50', $, #38, .T.);
#2696=PROPERTY_DEFINITION('', $, #2695);
#2697=FACE_SHAPE_REPRESENTATION('', (#2694), #29);
#2698=PROPERTY_DEFINITION_REPRESENTATION(#2696, #2697);
#2699=ORIENTED_EDGE('', *, *, #414, .T.);
#2700=CARTESIAN_POINT('', (196.0, -20.0, 27.500000000000000));
#2701=VERTEX_POINT('', #2700);
#2702=DIRECTION('', (0.0, 1.0, 0.0));
#2703=VECTOR('', #2702, 30.0);
#2704=LINE('', #407, #2703);
#2705=EDGE_CURVE('', #408, #2701, #2704, .T.);
#2706=ORIENTED_EDGE('', *, *, #2705, .T.);
#2707=CARTESIAN_POINT('', (214.0, -20.0, 27.500000000000000));
#2708=VERTEX_POINT('', #2707);
#2709=CARTESIAN_POINT('', (205.0, -20.0, 27.500000000000000));
#2710=DIRECTION('', (0.0, 1.0, 0.0));
#2711=DIRECTION('', (1.0, 0.0, 0.0));
#2712=AXIS2_PLACEMENT_3D('', #2709, #2710, #2711);
#2713=CIRCLE('', #2712, 9.0);
#2714=EDGE_CURVE('', #2708, #2701, #2713, .T.);
#2715=ORIENTED_EDGE('', *, *, #2714, .F.);
#2716=DIRECTION('', (0.0, 1.0, 0.0));
#2717=VECTOR('', #2716, 30.0);
#2718=LINE('', #405, #2717);
#2719=EDGE_CURVE('', #406, #2708, #2718, .T.);
#2720=ORIENTED_EDGE('', *, *, #2719, .F.);
#2721=EDGE_LOOP('', (#2699, #2706, #2715, #2720));
#2722=FACE_BOUND('', #2721, .F.);
#2723=CARTESIAN_POINT('', (205.0, -52.500000000000000, 27.500000000000000));

```

```

#2724=DIRECTION('', (0.0,1.0,0.0));
#2725=DIRECTION('', (1.0,0.0,0.0));
#2726=AXIS2_PLACEMENT_3D('', #2723, #2724, #2725);
#2727=CYLINDRICAL_SURFACE('', #2726, 9.0);
#2728=ADVANCED_FACE('', (#2722), #2727, .F.);
#2729=SHAPE_ASPECT('featured shape: face 51', $, #38, .T.);
#2730=PROPERTY_DEFINITION('', $, #2729);
#2731=FACE_SHAPE_REPRESENTATION('', (#2728), #29);
#2732=PROPERTY_DEFINITION_REPRESENTATION(#2730, #2731);
#2733=CARTESIAN_POINT('', (218.0, -20.0, 27.500000000000000));
#2734=VERTEX_POINT('', #2733);
#2735=CARTESIAN_POINT('', (192.0, -20.0, 27.500000000000000));
#2736=VERTEX_POINT('', #2735);
#2737=CARTESIAN_POINT('', (205.0, -20.0, 27.500000000000000));
#2738=DIRECTION('', (0.0,1.0,0.0));
#2739=DIRECTION('', (1.0,0.0,0.0));
#2740=AXIS2_PLACEMENT_3D('', #2737, #2738, #2739);
#2741=CIRCLE('', #2740, 13.0);
#2742=EDGE_CURVE('', #2734, #2736, #2741, .T.);
#2743=ORIENTED_EDGE('', *, *, #2742, .F.);
#2744=CARTESIAN_POINT('', (205.0, -20.0, 27.500000000000000));
#2745=DIRECTION('', (0.0, -1.0, 0.0));
#2746=DIRECTION('', (1.0, 0.0, 0.0));
#2747=AXIS2_PLACEMENT_3D('', #2744, #2745, #2746);
#2748=CIRCLE('', #2747, 13.0);
#2749=EDGE_CURVE('', #2734, #2736, #2748, .T.);
#2750=ORIENTED_EDGE('', *, *, #2749, .T.);
#2751=EDGE_LOOP('', (#2743, #2750));
#2752=FACE_BOUND('', #2751, .F.);
#2753=CARTESIAN_POINT('', (205.0, -20.0, 27.500000000000000));
#2754=DIRECTION('', (0.0, -1.0, 0.0));
#2755=DIRECTION('', (1.0, 0.0, 0.0));
#2756=AXIS2_PLACEMENT_3D('', #2753, #2754, #2755);
#2757=CIRCLE('', #2756, 9.0);
#2758=EDGE_CURVE('', #2708, #2701, #2757, .T.);
#2759=ORIENTED_EDGE('', *, *, #2758, .F.);
#2760=ORIENTED_EDGE('', *, *, #2714, .T.);
#2761=EDGE_LOOP('', (#2759, #2760));
#2762=FACE_BOUND('', #2761, .F.);
#2763=CARTESIAN_POINT('', (205.0, -20.0, 27.500000000000000));
#2764=DIRECTION('', (0.0,1.0,0.0));
#2765=DIRECTION('', (1.0,0.0,0.0));
#2766=AXIS2_PLACEMENT_3D('', #2763, #2764, #2765);
#2767=PLANE('', #2766);
#2768=ADVANCED_FACE('', (#2752, #2762), #2767, .T.);
#2769=SHAPE_ASPECT('featured shape: face 52', $, #38, .T.);
#2770=PROPERTY_DEFINITION('', $, #2769);
#2771=FACE_SHAPE_REPRESENTATION('', (#2768), #29);
#2772=PROPERTY_DEFINITION_REPRESENTATION(#2770, #2771);
#2773=ORIENTED_EDGE('', *, *, #853, .F.);
#2774=DIRECTION('', (0.0,1.0,0.0));
#2775=VECTOR('', #2774, 20.0);
#2776=LINE('', #2733, #2775);

```

```

#2777=EDGE_CURVE('',#2734,#845,#2776,.T.);
#2778=ORIENTED_EDGE('',*,*,#2777,.F.);
#2779=ORIENTED_EDGE('',*,*,#2742,.T.);
#2780=DIRECTION('',(0.0,1.0,0.0));
#2781=VECTOR('',#2780,20.0);
#2782=LINE('',#2735,#2781);
#2783=EDGE_CURVE('',#2736,#847,#2782,.T.);
#2784=ORIENTED_EDGE('',*,*,#2783,.T.);
#2785=EDGE_LOOP('',(#2773,#2778,#2779,#2784));
#2786=FACE_BOUND('',#2785,.F.);
#2787=CARTESIAN_POINT('',(205.0,-52.50000000000000,27.50000000000000));
#2788=DIRECTION('',(0.0,1.0,0.0));
#2789=DIRECTION('',(1.0,0.0,0.0));
#2790=AXIS2_PLACEMENT_3D('',#2787,#2788,#2789);
#2791=CYLINDRICAL_SURFACE('',#2790,13.0);
#2792=ADVANCED_FACE('',(#2786),#2791,.F.);
#2793=SHAPE_ASPECT('featured shape: face 53',$,#38,.T.);
#2794=PROPERTY_DEFINITION('',$,#2793);
#2795=FACE_SHAPE_REPRESENTATION('',(#2792),#29);
#2796=PROPERTY_DEFINITION_REPRESENTATION(#2794,#2795);
#2797=ORIENTED_EDGE('',*,*,#421,.T.);
#2798=ORIENTED_EDGE('',*,*,#2719,.T.);
#2799=ORIENTED_EDGE('',*,*,#2758,.T.);
#2800=ORIENTED_EDGE('',*,*,#2705,.F.);
#2801=EDGE_LOOP('',(#2797,#2798,#2799,#2800));
#2802=FACE_BOUND('',#2801,.F.);
#2803=CARTESIAN_POINT('',(205.0,-52.50000000000000,27.50000000000000));
#2804=DIRECTION('',(0.0,1.0,0.0));
#2805=DIRECTION('',(1.0,0.0,0.0));
#2806=AXIS2_PLACEMENT_3D('',#2803,#2804,#2805);
#2807=CYLINDRICAL_SURFACE('',#2806,9.0);
#2808=ADVANCED_FACE('',(#2802),#2807,.F.);
#2809=SHAPE_ASPECT('featured shape: face 54',$,#38,.T.);
#2810=PROPERTY_DEFINITION('',$,#2809);
#2811=FACE_SHAPE_REPRESENTATION('',(#2808),#29);
#2812=PROPERTY_DEFINITION_REPRESENTATION(#2810,#2811);
#2813=ORIENTED_EDGE('',*,*,#860,.F.);
#2814=ORIENTED_EDGE('',*,*,#2783,.F.);
#2815=ORIENTED_EDGE('',*,*,#2749,.F.);
#2816=ORIENTED_EDGE('',*,*,#2777,.T.);
#2817=EDGE_LOOP('',(#2813,#2814,#2815,#2816));
#2818=FACE_BOUND('',#2817,.F.);
#2819=CARTESIAN_POINT('',(205.0,-52.50000000000000,27.50000000000000));
#2820=DIRECTION('',(0.0,1.0,0.0));
#2821=DIRECTION('',(1.0,0.0,0.0));
#2822=AXIS2_PLACEMENT_3D('',#2819,#2820,#2821);
#2823=CYLINDRICAL_SURFACE('',#2822,13.0);
#2824=ADVANCED_FACE('',(#2818),#2823,.F.);
#2825=SHAPE_ASPECT('featured shape: face 55',$,#38,.T.);
#2826=PROPERTY_DEFINITION('',$,#2825);
#2827=FACE_SHAPE_REPRESENTATION('',(#2824),#29);
#2828=PROPERTY_DEFINITION_REPRESENTATION(#2826,#2827);
#2829=ORIENTED_EDGE('',*,*,#505,.T.);

```

```

#2830=CARTESIAN_POINT('', (196.0, -20.0, 152.50000000000000));
#2831=VERTEX_POINT('', #2830);
#2832=DIRECTION('', (0.0, 1.0, 0.0));
#2833=VECTOR('', #2832, 30.0);
#2834=LINE('', #498, #2833);
#2835=EDGE_CURVE('', #499, #2831, #2834, .T.);
#2836=ORIENTED_EDGE('', *, *, #2835, .T.);
#2837=CARTESIAN_POINT('', (214.0, -20.0, 152.50000000000000));
#2838=VERTEX_POINT('', #2837);
#2839=CARTESIAN_POINT('', (205.0, -20.0, 152.50000000000000));
#2840=DIRECTION('', (0.0, 1.0, 0.0));
#2841=DIRECTION('', (1.0, 0.0, 0.0));
#2842=AXIS2_PLACEMENT_3D('', #2839, #2840, #2841);
#2843=CIRCLE('', #2842, 9.0);
#2844=EDGE_CURVE('', #2838, #2831, #2843, .T.);
#2845=ORIENTED_EDGE('', *, *, #2844, .F.);
#2846=DIRECTION('', (0.0, 1.0, 0.0));
#2847=VECTOR('', #2846, 30.0);
#2848=LINE('', #496, #2847);
#2849=EDGE_CURVE('', #497, #2838, #2848, .T.);
#2850=ORIENTED_EDGE('', *, *, #2849, .F.);
#2851=EDGE_LOOP('', (#2829, #2836, #2845, #2850));
#2852=FACE_BOUND('', #2851, .F.);
#2853=CARTESIAN_POINT('', (205.0, -52.50000000000000, 152.50000000000000));
#2854=DIRECTION('', (0.0, 1.0, 0.0));
#2855=DIRECTION('', (1.0, 0.0, 0.0));
#2856=AXIS2_PLACEMENT_3D('', #2853, #2854, #2855);
#2857=CYLINDRICAL_SURFACE('', #2856, 9.0);
#2858=ADVANCED_FACE('', (#2852), #2857, .F.);
#2859=SHAPE_ASPECT('featured shape: face 56', $, #38, .T.);
#2860=PROPERTY_DEFINITION('', $, #2859);
#2861=FACE_SHAPE_REPRESENTATION('', (#2858), #29);
#2862=PROPERTY_DEFINITION_REPRESENTATION(#2860, #2861);
#2863=CARTESIAN_POINT('', (218.0, -20.0, 152.50000000000000));
#2864=VERTEX_POINT('', #2863);
#2865=CARTESIAN_POINT('', (192.0, -20.0, 152.50000000000000));
#2866=VERTEX_POINT('', #2865);
#2867=CARTESIAN_POINT('', (205.0, -20.0, 152.50000000000000));
#2868=DIRECTION('', (0.0, 1.0, 0.0));
#2869=DIRECTION('', (1.0, 0.0, 0.0));
#2870=AXIS2_PLACEMENT_3D('', #2867, #2868, #2869);
#2871=CIRCLE('', #2870, 13.0);
#2872=EDGE_CURVE('', #2864, #2866, #2871, .T.);
#2873=ORIENTED_EDGE('', *, *, #2872, .F.);
#2874=CARTESIAN_POINT('', (205.0, -20.0, 152.50000000000000));
#2875=DIRECTION('', (0.0, -1.0, 0.0));
#2876=DIRECTION('', (1.0, 0.0, 0.0));
#2877=AXIS2_PLACEMENT_3D('', #2874, #2875, #2876);
#2878=CIRCLE('', #2877, 13.0);
#2879=EDGE_CURVE('', #2864, #2866, #2878, .T.);
#2880=ORIENTED_EDGE('', *, *, #2879, .T.);
#2881=EDGE_LOOP('', (#2873, #2880));
#2882=FACE_BOUND('', #2881, .F.);

```

```

#2883=CARTESIAN_POINT('', (205.0, -20.0, 152.50000000000000));
#2884=DIRECTION('', (0.0, -1.0, 0.0));
#2885=DIRECTION('', (1.0, 0.0, 0.0));
#2886=AXIS2_PLACEMENT_3D('', #2883, #2884, #2885);
#2887=CIRCLE('', #2886, 9.0);
#2888=EDGE_CURVE('', #2838, #2831, #2887, .T.);
#2889=ORIENTED_EDGE('', *, *, #2888, .F.);
#2890=ORIENTED_EDGE('', *, *, #2844, .T.);
#2891=EDGE_LOOP('', (#2889, #2890));
#2892=FACE_BOUND('', #2891, .F.);
#2893=CARTESIAN_POINT('', (205.0, -20.0, 152.50000000000000));
#2894=DIRECTION('', (0.0, 1.0, 0.0));
#2895=DIRECTION('', (1.0, 0.0, 0.0));
#2896=AXIS2_PLACEMENT_3D('', #2893, #2894, #2895);
#2897=PLANE('', #2896);
#2898=ADVANCED_FACE('', (#2882, #2892), #2897, .T.);
#2899=SHAPE_ASPECT('featured shape: face 57', $, #38, .T.);
#2900=PROPERTY_DEFINITION('', $, #2899);
#2901=FACE_SHAPE_REPRESENTATION('', (#2898), #29);
#2902=PROPERTY_DEFINITION_REPRESENTATION(#2900, #2901);
#2903=ORIENTED_EDGE('', *, *, #873, .F.);
#2904=DIRECTION('', (0.0, 1.0, 0.0));
#2905=VECTOR('', #2904, 20.0);
#2906=LINE('', #2863, #2905);
#2907=EDGE_CURVE('', #2864, #865, #2906, .T.);
#2908=ORIENTED_EDGE('', *, *, #2907, .F.);
#2909=ORIENTED_EDGE('', *, *, #2872, .T.);
#2910=DIRECTION('', (0.0, 1.0, 0.0));
#2911=VECTOR('', #2910, 20.0);
#2912=LINE('', #2865, #2911);
#2913=EDGE_CURVE('', #2866, #867, #2912, .T.);
#2914=ORIENTED_EDGE('', *, *, #2913, .T.);
#2915=EDGE_LOOP('', (#2903, #2908, #2909, #2914));
#2916=FACE_BOUND('', #2915, .F.);
#2917=CARTESIAN_POINT('', (205.0, -52.50000000000000, 152.50000000000000));
#2918=DIRECTION('', (0.0, 1.0, 0.0));
#2919=DIRECTION('', (1.0, 0.0, 0.0));
#2920=AXIS2_PLACEMENT_3D('', #2917, #2918, #2919);
#2921=CYLINDRICAL_SURFACE('', #2920, 13.0);
#2922=ADVANCED_FACE('', (#2916), #2921, .F.);
#2923=SHAPE_ASPECT('featured shape: face 58', $, #38, .T.);
#2924=PROPERTY_DEFINITION('', $, #2923);
#2925=FACE_SHAPE_REPRESENTATION('', (#2922), #29);
#2926=PROPERTY_DEFINITION_REPRESENTATION(#2924, #2925);
#2927=ORIENTED_EDGE('', *, *, #512, .T.);
#2928=ORIENTED_EDGE('', *, *, #2849, .T.);
#2929=ORIENTED_EDGE('', *, *, #2888, .T.);
#2930=ORIENTED_EDGE('', *, *, #2835, .F.);
#2931=EDGE_LOOP('', (#2927, #2928, #2929, #2930));
#2932=FACE_BOUND('', #2931, .F.);
#2933=CARTESIAN_POINT('', (205.0, -52.50000000000000, 152.50000000000000));
#2934=DIRECTION('', (0.0, 1.0, 0.0));
#2935=DIRECTION('', (1.0, 0.0, 0.0));

```

ISO 10303-223:2008 (E)

```

#2936=AXIS2_PLACEMENT_3D('',#2933,#2934,#2935);
#2937=CYLINDRICAL_SURFACE('',#2936,9.0);
#2938=ADVANCED_FACE('',(#2932),#2937,.F.);
#2939=SHAPE_ASPECT('featured shape: face 59',$,#38,.T.);
#2940=PROPERTY_DEFINITION('',$,#2939);
#2941=FACE_SHAPE_REPRESENTATION('',(#2938),#29);
#2942=PROPERTY_DEFINITION_REPRESENTATION(#2940,#2941);
#2943=ORIENTED_EDGE('',*,*,#880,.F.);
#2944=ORIENTED_EDGE('',*,*,#2913,.F.);
#2945=ORIENTED_EDGE('',*,*,#2879,.F.);
#2946=ORIENTED_EDGE('',*,*,#2907,.T.);
#2947=EDGE_LOOP('',(#2943,#2944,#2945,#2946));
#2948=FACE_BOUND('',#2947,.F.);
#2949=CARTESIAN_POINT('',(205.0,-52.50000000000000,152.50000000000000));
#2950=DIRECTION('',(0.0,1.0,0.0));
#2951=DIRECTION('',(1.0,0.0,0.0));
#2952=AXIS2_PLACEMENT_3D('',#2949,#2950,#2951);
#2953=CYLINDRICAL_SURFACE('',#2952,13.0);
#2954=ADVANCED_FACE('',(#2948),#2953,.F.);
#2955=SHAPE_ASPECT('featured shape: face 60',$,#38,.T.);
#2956=PROPERTY_DEFINITION('',$,#2955);
#2957=FACE_SHAPE_REPRESENTATION('',(#2954),#29);
#2958=PROPERTY_DEFINITION_REPRESENTATION(#2956,#2957);
#2959=ORIENTED_EDGE('',*,*,#318,.T.);
#2960=CARTESIAN_POINT('',(16.0,-20.0,27.50000000000000));
#2961=VERTEX_POINT('',#2960);
#2962=DIRECTION('',(0.0,1.0,0.0));
#2963=VECTOR('',#2962,30.0);
#2964=LINE('',#311,#2963);
#2965=EDGE_CURVE('',#312,#2961,#2964,.T.);
#2966=ORIENTED_EDGE('',*,*,#2965,.T.);
#2967=CARTESIAN_POINT('',(34.0,-20.0,27.50000000000000));
#2968=VERTEX_POINT('',#2967);
#2969=CARTESIAN_POINT('',(25.0,-20.0,27.50000000000000));
#2970=DIRECTION('',(0.0,1.0,0.0));
#2971=DIRECTION('',(1.0,0.0,0.0));
#2972=AXIS2_PLACEMENT_3D('',#2969,#2970,#2971);
#2973=CIRCLE('',#2972,9.0);
#2974=EDGE_CURVE('',#2968,#2961,#2973,.T.);
#2975=ORIENTED_EDGE('',*,*,#2974,.F.);
#2976=DIRECTION('',(0.0,1.0,0.0));
#2977=VECTOR('',#2976,30.0);
#2978=LINE('',#309,#2977);
#2979=EDGE_CURVE('',#310,#2968,#2978,.T.);
#2980=ORIENTED_EDGE('',*,*,#2979,.F.);
#2981=EDGE_LOOP('',(#2959,#2966,#2975,#2980));
#2982=FACE_BOUND('',#2981,.F.);
#2983=CARTESIAN_POINT('',(25.0,-52.50000000000000,27.50000000000000));
#2984=DIRECTION('',(0.0,1.0,0.0));
#2985=DIRECTION('',(1.0,0.0,0.0));
#2986=AXIS2_PLACEMENT_3D('',#2983,#2984,#2985);
#2987=CYLINDRICAL_SURFACE('',#2986,9.0);
#2988=ADVANCED_FACE('',(#2982),#2987,.F.);

```



```

#2989=SHAPE_ASPECT('featured shape: face 61',$, #38, .T.);
#2990=PROPERTY_DEFINITION('', $, #2989);
#2991=FACE_SHAPE_REPRESENTATION('', (#2988), #29);
#2992=PROPERTY_DEFINITION_REPRESENTATION(#2990, #2991);
#2993=CARTESIAN_POINT('', (38.0, -20.0, 27.500000000000000));
#2994=VERTEX_POINT('', #2993);
#2995=CARTESIAN_POINT('', (12.0, -20.0, 27.500000000000000));
#2996=VERTEX_POINT('', #2995);
#2997=CARTESIAN_POINT('', (25.0, -20.0, 27.500000000000000));
#2998=DIRECTION('', (0.0, 1.0, 0.0));
#2999=DIRECTION('', (1.0, 0.0, 0.0));
#3000=AXIS2_PLACEMENT_3D('', #2997, #2998, #2999);
#3001=CIRCLE('', #3000, 13.0);
#3002=EDGE_CURVE('', #2994, #2996, #3001, .T.);
#3003=ORIENTED_EDGE('', *, *, #3002, .F.);
#3004=CARTESIAN_POINT('', (25.0, -20.0, 27.500000000000000));
#3005=DIRECTION('', (0.0, -1.0, 0.0));
#3006=DIRECTION('', (1.0, 0.0, 0.0));
#3007=AXIS2_PLACEMENT_3D('', #3004, #3005, #3006);
#3008=CIRCLE('', #3007, 13.0);
#3009=EDGE_CURVE('', #2994, #2996, #3008, .T.);
#3010=ORIENTED_EDGE('', *, *, #3009, .T.);
#3011=EDGE_LOOP('', (#3003, #3010));
#3012=FACE_BOUND('', #3011, .F.);
#3013=CARTESIAN_POINT('', (25.0, -20.0, 27.500000000000000));
#3014=DIRECTION('', (0.0, -1.0, 0.0));
#3015=DIRECTION('', (1.0, 0.0, 0.0));
#3016=AXIS2_PLACEMENT_3D('', #3013, #3014, #3015);
#3017=CIRCLE('', #3016, 9.0);
#3018=EDGE_CURVE('', #2968, #2961, #3017, .T.);
#3019=ORIENTED_EDGE('', *, *, #3018, .F.);
#3020=ORIENTED_EDGE('', *, *, #2974, .T.);
#3021=EDGE_LOOP('', (#3019, #3020));
#3022=FACE_BOUND('', #3021, .F.);
#3023=CARTESIAN_POINT('', (25.0, -20.0, 27.500000000000000));
#3024=DIRECTION('', (0.0, 1.0, 0.0));
#3025=DIRECTION('', (1.0, 0.0, 0.0));
#3026=AXIS2_PLACEMENT_3D('', #3023, #3024, #3025);
#3027=PLANE('', #3026);
#3028=ADVANCED_FACE('', (#3012, #3022), #3027, .T.);
#3029=SHAPE_ASPECT('featured shape: face 62', $, #38, .T.);
#3030=PROPERTY_DEFINITION('', $, #3029);
#3031=FACE_SHAPE_REPRESENTATION('', (#3028), #29);
#3032=PROPERTY_DEFINITION_REPRESENTATION(#3030, #3031);
#3033=ORIENTED_EDGE('', *, *, #893, .F.);
#3034=DIRECTION('', (0.0, 1.0, 0.0));
#3035=VECTOR('', #3034, 20.0);
#3036=LINE('', #2993, #3035);
#3037=EDGE_CURVE('', #2994, #885, #3036, .T.);
#3038=ORIENTED_EDGE('', *, *, #3037, .F.);
#3039=ORIENTED_EDGE('', *, *, #3002, .T.);
#3040=DIRECTION('', (0.0, 1.0, 0.0));
#3041=VECTOR('', #3040, 20.0);

```

```

#3042=LINE('',#2995,#3041);
#3043=EDGE_CURVE('',#2996,#887,#3042,.T.);
#3044=ORIENTED_EDGE('',*,*,#3043,.T.);
#3045=EDGE_LOOP('',(#3033,#3038,#3039,#3044));
#3046=FACE_BOUND('',#3045,.F.);
#3047=CARTESIAN_POINT('',(25.0,-52.50000000000000,27.50000000000000));
#3048=DIRECTION('',(0.0,1.0,0.0));
#3049=DIRECTION('',(1.0,0.0,0.0));
#3050=AXIS2_PLACEMENT_3D('',#3047,#3048,#3049);
#3051=CYLINDRICAL_SURFACE('',#3050,13.0);
#3052=ADVANCED_FACE('',(#3046),#3051,.F.);
#3053=SHAPE_ASPECT('featured shape: face 63',$,#38,.T.);
#3054=PROPERTY_DEFINITION('',$,#3053);
#3055=FACE_SHAPE_REPRESENTATION('',(#3052),#29);
#3056=PROPERTY_DEFINITION_REPRESENTATION(#3054,#3055);
#3057=ORIENTED_EDGE('',*,*,#325,.T.);
#3058=ORIENTED_EDGE('',*,*,#2979,.T.);
#3059=ORIENTED_EDGE('',*,*,#3018,.T.);
#3060=ORIENTED_EDGE('',*,*,#2965,.F.);
#3061=EDGE_LOOP('',(#3057,#3058,#3059,#3060));
#3062=FACE_BOUND('',#3061,.F.);
#3063=CARTESIAN_POINT('',(25.0,-52.50000000000000,27.50000000000000));
#3064=DIRECTION('',(0.0,1.0,0.0));
#3065=DIRECTION('',(1.0,0.0,0.0));
#3066=AXIS2_PLACEMENT_3D('',#3063,#3064,#3065);
#3067=CYLINDRICAL_SURFACE('',#3066,9.0);
#3068=ADVANCED_FACE('',(#3062),#3067,.F.);
#3069=SHAPE_ASPECT('featured shape: face 64',$,#38,.T.);
#3070=PROPERTY_DEFINITION('',$,#3069);
#3071=FACE_SHAPE_REPRESENTATION('',(#3068),#29);
#3072=PROPERTY_DEFINITION_REPRESENTATION(#3070,#3071);
#3073=ORIENTED_EDGE('',*,*,#900,.F.);
#3074=ORIENTED_EDGE('',*,*,#3043,.F.);
#3075=ORIENTED_EDGE('',*,*,#3009,.F.);
#3076=ORIENTED_EDGE('',*,*,#3037,.T.);
#3077=EDGE_LOOP('',(#3073,#3074,#3075,#3076));
#3078=FACE_BOUND('',#3077,.F.);
#3079=CARTESIAN_POINT('',(25.0,-52.50000000000000,27.50000000000000));
#3080=DIRECTION('',(0.0,1.0,0.0));
#3081=DIRECTION('',(1.0,0.0,0.0));
#3082=AXIS2_PLACEMENT_3D('',#3079,#3080,#3081);
#3083=CYLINDRICAL_SURFACE('',#3082,13.0);
#3084=ADVANCED_FACE('',(#3078),#3083,.F.);
#3085=SHAPE_ASPECT('featured shape: face 65',$,#38,.T.);
#3086=PROPERTY_DEFINITION('',$,#3085);
#3087=FACE_SHAPE_REPRESENTATION('',(#3084),#29);
#3088=PROPERTY_DEFINITION_REPRESENTATION(#3086,#3087);
#3089=ORIENTED_EDGE('',*,*,#596,.T.);
#3090=CARTESIAN_POINT('',(16.0,-20.0,152.50000000000000));
#3091=VERTEX_POINT('',#3090);
#3092=DIRECTION('',(0.0,1.0,0.0));
#3093=VECTOR('',#3092,30.0);
#3094=LINE('',#589,#3093);

```

```

#3095=EDGE_CURVE('',#590,#3091,#3094,.T.);
#3096=ORIENTED_EDGE('',*,*,#3095,.T.);
#3097=CARTESIAN_POINT('',(34.0,-20.0,152.50000000000000));
#3098=VERTEX_POINT('',#3097);
#3099=CARTESIAN_POINT('',(25.0,-20.0,152.50000000000000));
#3100=DIRECTION('',(0.0,1.0,0.0));
#3101=DIRECTION('',(1.0,0.0,0.0));
#3102=AXIS2_PLACEMENT_3D('',#3099,#3100,#3101);
#3103=CIRCLE('',#3102,9.0);
#3104=EDGE_CURVE('',#3098,#3091,#3103,.T.);
#3105=ORIENTED_EDGE('',*,*,#3104,.F.);
#3106=DIRECTION('',(0.0,1.0,0.0));
#3107=VECTOR('',#3106,30.0);
#3108=LINE('',#587,#3107);
#3109=EDGE_CURVE('',#588,#3098,#3108,.T.);
#3110=ORIENTED_EDGE('',*,*,#3109,.F.);
#3111=EDGE_LOOP('',(#3089,#3096,#3105,#3110));
#3112=FACE_BOUND('',#3111,.F.);
#3113=CARTESIAN_POINT('',(25.0,-52.50000000000000,152.50000000000000));
#3114=DIRECTION('',(0.0,1.0,0.0));
#3115=DIRECTION('',(1.0,0.0,0.0));
#3116=AXIS2_PLACEMENT_3D('',#3113,#3114,#3115);
#3117=CYLINDRICAL_SURFACE('',#3116,9.0);
#3118=ADVANCED_FACE('',(#3112),#3117,.F.);
#3119=SHAPE_ASPECT('featured shape: face 66',$,#38,.T.);
#3120=PROPERTY_DEFINITION('',$,#3119);
#3121=FACE_SHAPE_REPRESENTATION('',(#3118),#29);
#3122=PROPERTY_DEFINITION_REPRESENTATION(#3120,#3121);
#3123=CARTESIAN_POINT('',(38.0,-20.0,152.50000000000000));
#3124=VERTEX_POINT('',#3123);
#3125=CARTESIAN_POINT('',(12.0,-20.0,152.50000000000000));
#3126=VERTEX_POINT('',#3125);
#3127=CARTESIAN_POINT('',(25.0,-20.0,152.50000000000000));
#3128=DIRECTION('',(0.0,1.0,0.0));
#3129=DIRECTION('',(1.0,0.0,0.0));
#3130=AXIS2_PLACEMENT_3D('',#3127,#3128,#3129);
#3131=CIRCLE('',#3130,13.0);
#3132=EDGE_CURVE('',#3124,#3126,#3131,.T.);
#3133=ORIENTED_EDGE('',*,*,#3132,.F.);
#3134=CARTESIAN_POINT('',(25.0,-20.0,152.50000000000000));
#3135=DIRECTION('',(0.0,-1.0,0.0));
#3136=DIRECTION('',(1.0,0.0,0.0));
#3137=AXIS2_PLACEMENT_3D('',#3134,#3135,#3136);
#3138=CIRCLE('',#3137,13.0);
#3139=EDGE_CURVE('',#3124,#3126,#3138,.T.);
#3140=ORIENTED_EDGE('',*,*,#3139,.T.);
#3141=EDGE_LOOP('',(#3133,#3140));
#3142=FACE_BOUND('',#3141,.F.);
#3143=CARTESIAN_POINT('',(25.0,-20.0,152.50000000000000));
#3144=DIRECTION('',(0.0,-1.0,0.0));
#3145=DIRECTION('',(1.0,0.0,0.0));
#3146=AXIS2_PLACEMENT_3D('',#3143,#3144,#3145);
#3147=CIRCLE('',#3146,9.0);

```

```

#3148=EDGE_CURVE('',#3098,#3091,#3147,.T.);
#3149=ORIENTED_EDGE('',*,*,#3148,.F.);
#3150=ORIENTED_EDGE('',*,*,#3104,.T.);
#3151=EDGE_LOOP('',(#3149,#3150));
#3152=FACE_BOUND('',#3151,.F.);
#3153=CARTESIAN_POINT('',(25.0,-20.0,152.50000000000000));
#3154=DIRECTION('',(0.0,1.0,0.0));
#3155=DIRECTION('',(1.0,0.0,0.0));
#3156=AXIS2_PLACEMENT_3D('',#3153,#3154,#3155);
#3157=PLANE('',#3156);
#3158=ADVANCED_FACE('',(#3142,#3152),#3157,.T.);
#3159=SHAPE_ASPECT('featured shape: face 67',$,#38,.T.);
#3160=PROPERTY_DEFINITION('',$,#3159);
#3161=FACE_SHAPE_REPRESENTATION('',(#3158),#29);
#3162=PROPERTY_DEFINITION_REPRESENTATION(#3160,#3161);
#3163=ORIENTED_EDGE('',*,*,#913,.F.);
#3164=DIRECTION('',(0.0,1.0,0.0));
#3165=VECTOR('',#3164,20.0);
#3166=LINE('',#3123,#3165);
#3167=EDGE_CURVE('',#3124,#905,#3166,.T.);
#3168=ORIENTED_EDGE('',*,*,#3167,.F.);
#3169=ORIENTED_EDGE('',*,*,#3132,.T.);
#3170=DIRECTION('',(0.0,1.0,0.0));
#3171=VECTOR('',#3170,20.0);
#3172=LINE('',#3125,#3171);
#3173=EDGE_CURVE('',#3126,#907,#3172,.T.);
#3174=ORIENTED_EDGE('',*,*,#3173,.T.);
#3175=EDGE_LOOP('',(#3163,#3168,#3169,#3174));
#3176=FACE_BOUND('',#3175,.F.);
#3177=CARTESIAN_POINT('',(25.0,-52.50000000000000,152.50000000000000));
#3178=DIRECTION('',(0.0,1.0,0.0));
#3179=DIRECTION('',(1.0,0.0,0.0));
#3180=AXIS2_PLACEMENT_3D('',#3177,#3178,#3179);
#3181=CYLINDRICAL_SURFACE('',#3180,13.0);
#3182=ADVANCED_FACE('',(#3176),#3181,.F.);
#3183=SHAPE_ASPECT('featured shape: face 68',$,#38,.T.);
#3184=PROPERTY_DEFINITION('',$,#3183);
#3185=FACE_SHAPE_REPRESENTATION('',(#3182),#29);
#3186=PROPERTY_DEFINITION_REPRESENTATION(#3184,#3185);
#3187=ORIENTED_EDGE('',*,*,#603,.T.);
#3188=ORIENTED_EDGE('',*,*,#3109,.T.);
#3189=ORIENTED_EDGE('',*,*,#3148,.T.);
#3190=ORIENTED_EDGE('',*,*,#3095,.F.);
#3191=EDGE_LOOP('',(#3187,#3188,#3189,#3190));
#3192=FACE_BOUND('',#3191,.F.);
#3193=CARTESIAN_POINT('',(25.0,-52.50000000000000,152.50000000000000));
#3194=DIRECTION('',(0.0,1.0,0.0));
#3195=DIRECTION('',(1.0,0.0,0.0));
#3196=AXIS2_PLACEMENT_3D('',#3193,#3194,#3195);
#3197=CYLINDRICAL_SURFACE('',#3196,9.0);
#3198=ADVANCED_FACE('',(#3192),#3197,.F.);
#3199=SHAPE_ASPECT('featured shape: face 69',$,#38,.T.);
#3200=PROPERTY_DEFINITION('',$,#3199);

```

```

#3201=FACE_SHAPE_REPRESENTATION('', (#3198), #29);
#3202=PROPERTY_DEFINITION_REPRESENTATION(#3200, #3201);
#3203=ORIENTED_EDGE('', *, *, #920, .F.);
#3204=ORIENTED_EDGE('', *, *, #3173, .F.);
#3205=ORIENTED_EDGE('', *, *, #3139, .F.);
#3206=ORIENTED_EDGE('', *, *, #3167, .T.);
#3207=EDGE_LOOP('', (#3203, #3204, #3205, #3206));
#3208=FACE_BOUND('', #3207, .F.);
#3209=CARTESIAN_POINT('', (25.0, -52.50000000000000, 152.50000000000000));
#3210=DIRECTION('', (0.0, 1.0, 0.0));
#3211=DIRECTION('', (1.0, 0.0, 0.0));
#3212=AXIS2_PLACEMENT_3D('', #3209, #3210, #3211);
#3213=CYLINDRICAL_SURFACE('', #3212, 13.0);
#3214=ADVANCED_FACE('', (#3208), #3213, .F.);
#3215=SHAPE_ASPECT('featured shape: face 70', $, #38, .T.);
#3216=PROPERTY_DEFINITION('', $, #3215);
#3217=FACE_SHAPE_REPRESENTATION('', (#3214), #29);
#3218=PROPERTY_DEFINITION_REPRESENTATION(#3216, #3217);
#3219=ORIENTED_EDGE('', *, *, #765, .T.);
#3220=CARTESIAN_POINT('', (210.0, -10.0, 95.0));
#3221=VERTEX_POINT('', #3220);
#3222=DIRECTION('', (0.0, -1.0, 0.0));
#3223=VECTOR('', #3222, 10.0);
#3224=LINE('', #751, #3223);
#3225=EDGE_CURVE('', #752, #3221, #3224, .T.);
#3226=ORIENTED_EDGE('', *, *, #3225, .T.);
#3227=DIRECTION('', (1.0, 0.0, 0.0));
#3228=VECTOR('', #3227, 20.0);
#3229=LINE('', #3220, #3228);
#3230=EDGE_CURVE('', #3221, #1186, #3229, .T.);
#3231=ORIENTED_EDGE('', *, *, #3230, .T.);
#3232=ORIENTED_EDGE('', *, *, #1190, .F.);
#3233=EDGE_LOOP('', (#3219, #3226, #3231, #3232));
#3234=FACE_BOUND('', #3233, .F.);
#3235=CARTESIAN_POINT('', (205.0, 0.0, 95.0));
#3236=DIRECTION('', (0.0, 0.0, 1.0));
#3237=DIRECTION('', (0.0, -1.0, 0.0));
#3238=AXIS2_PLACEMENT_3D('', #3235, #3236, #3237);
#3239=PLANE('', #3238);
#3240=ADVANCED_FACE('', (#3234), #3239, .F.);
#3241=SHAPE_ASPECT('featured shape: face 71', $, #38, .T.);
#3242=PROPERTY_DEFINITION('', $, #3241);
#3243=FACE_SHAPE_REPRESENTATION('', (#3240), #29);
#3244=PROPERTY_DEFINITION_REPRESENTATION(#3242, #3243);
#3245=ORIENTED_EDGE('', *, *, #1197, .F.);
#3246=ORIENTED_EDGE('', *, *, #3230, .F.);
#3247=CARTESIAN_POINT('', (210.0, -10.0, 85.0));
#3248=VERTEX_POINT('', #3247);
#3249=CARTESIAN_POINT('', (210.0, -10.0, 90.0));
#3250=DIRECTION('', (0.0, 1.0, 0.0));
#3251=DIRECTION('', (0.0, 0.0, -1.0));
#3252=AXIS2_PLACEMENT_3D('', #3249, #3250, #3251);
#3253=CIRCLE('', #3252, 5.0);

```

ISO 10303-223:2008 (E)

```

#3254=EDGE_CURVE('',#3248,#3221,#3253,.T.);
#3255=ORIENTED_EDGE('',*,*,#3254,.F.);
#3256=DIRECTION('',(1.0,0.0,0.0));
#3257=VECTOR('',#3256,20.0);
#3258=LINE('',#3247,#3257);
#3259=EDGE_CURVE('',#3248,#1193,#3258,.T.);
#3260=ORIENTED_EDGE('',*,*,#3259,.T.);
#3261=EDGE_LOOP('',(#3245,#3246,#3255,#3260));
#3262=FACE_BOUND('',#3261,.F.);
#3263=CARTESIAN_POINT('',(205.0,-10.0,95.0));
#3264=DIRECTION('',(0.0,-1.0,0.0));
#3265=DIRECTION('',(0.0,0.0,-1.0));
#3266=AXIS2_PLACEMENT_3D('',#3263,#3264,#3265);
#3267=PLANE('',#3266);
#3268=ADVANCED_FACE('',(#3262),#3267,.F.);
#3269=SHAPE_ASPECT('featured shape: face 72',$,#38,.T.);
#3270=PROPERTY_DEFINITION('',$,#3269);
#3271=FACE_SHAPE_REPRESENTATION('',(#3268),#29);
#3272=PROPERTY_DEFINITION_REPRESENTATION(#3270,#3271);
#3273=ORIENTED_EDGE('',*,*,#749,.F.);
#3274=ORIENTED_EDGE('',*,*,#1202,.T.);
#3275=ORIENTED_EDGE('',*,*,#3259,.F.);
#3276=DIRECTION('',(0.0,1.0,0.0));
#3277=VECTOR('',#3276,10.0);
#3278=LINE('',#3247,#3277);
#3279=EDGE_CURVE('',#3248,#745,#3278,.T.);
#3280=ORIENTED_EDGE('',*,*,#3279,.T.);
#3281=EDGE_LOOP('',(#3273,#3274,#3275,#3280));
#3282=FACE_BOUND('',#3281,.F.);
#3283=CARTESIAN_POINT('',(205.0,-10.0,85.0));
#3284=DIRECTION('',(0.0,0.0,-1.0));
#3285=DIRECTION('',(0.0,1.0,0.0));
#3286=AXIS2_PLACEMENT_3D('',#3283,#3284,#3285);
#3287=PLANE('',#3286);
#3288=ADVANCED_FACE('',(#3282),#3287,.F.);
#3289=SHAPE_ASPECT('featured shape: face 73',$,#38,.T.);
#3290=PROPERTY_DEFINITION('',$,#3289);
#3291=FACE_SHAPE_REPRESENTATION('',(#3288),#29);
#3292=PROPERTY_DEFINITION_REPRESENTATION(#3290,#3291);
#3293=ORIENTED_EDGE('',*,*,#3254,.T.);
#3294=ORIENTED_EDGE('',*,*,#3225,.F.);
#3295=ORIENTED_EDGE('',*,*,#758,.T.);
#3296=ORIENTED_EDGE('',*,*,#3279,.F.);
#3297=EDGE_LOOP('',(#3293,#3294,#3295,#3296));
#3298=FACE_BOUND('',#3297,.F.);
#3299=CARTESIAN_POINT('',(210.0,-10.0,90.0));
#3300=DIRECTION('',(0.0,-1.0,0.0));
#3301=DIRECTION('',(-1.0,0.0,0.0));
#3302=AXIS2_PLACEMENT_3D('',#3299,#3300,#3301);
#3303=CYLINDRICAL_SURFACE('',#3302,5.0);
#3304=ADVANCED_FACE('',(#3298),#3303,.F.);
#3305=SHAPE_ASPECT('featured shape: face 74',$,#38,.T.);
#3306=PROPERTY_DEFINITION('',$,#3305);

```

```

#3307=FACE_SHAPE_REPRESENTATION('', (#3304), #29);
#3308=PROPERTY_DEFINITION_REPRESENTATION(#3306, #3307);
#3309=CARTESIAN_POINT('', (110.0, -10.0, 85.0));
#3310=VERTEX_POINT('', #3309);
#3311=CARTESIAN_POINT('', (85.0, -10.0, 85.0));
#3312=VERTEX_POINT('', #3311);
#3313=DIRECTION('', (-1.0, 0.0, 0.0));
#3314=VECTOR('', #3313, 25.0);
#3315=LINE('', #3309, #3314);
#3316=EDGE_CURVE('', #3310, #3312, #3315, .T.);
#3317=ORIENTED_EDGE('', *, *, #3316, .F.);
#3318=CARTESIAN_POINT('', (110.0, -10.0, 95.0));
#3319=VERTEX_POINT('', #3318);
#3320=CARTESIAN_POINT('', (110.0, -10.0, 90.0));
#3321=DIRECTION('', (0.0, 1.0, 0.0));
#3322=DIRECTION('', (0.0, 0.0, 1.0));
#3323=AXIS2_PLACEMENT_3D('', #3320, #3321, #3322);
#3324=CIRCLE('', #3323, 5.0);
#3325=EDGE_CURVE('', #3319, #3310, #3324, .T.);
#3326=ORIENTED_EDGE('', *, *, #3325, .F.);
#3327=CARTESIAN_POINT('', (85.0, -10.0, 95.0));
#3328=VERTEX_POINT('', #3327);
#3329=DIRECTION('', (1.0, 0.0, 0.0));
#3330=VECTOR('', #3329, 25.0);
#3331=LINE('', #3327, #3330);
#3332=EDGE_CURVE('', #3328, #3319, #3331, .T.);
#3333=ORIENTED_EDGE('', *, *, #3332, .F.);
#3334=CARTESIAN_POINT('', (85.0, -10.0, 90.0));
#3335=DIRECTION('', (0.0, 1.0, 0.0));
#3336=DIRECTION('', (0.0, 0.0, -1.0));
#3337=AXIS2_PLACEMENT_3D('', #3334, #3335, #3336);
#3338=CIRCLE('', #3337, 5.0);
#3339=EDGE_CURVE('', #3312, #3328, #3338, .T.);
#3340=ORIENTED_EDGE('', *, *, #3339, .F.);
#3341=EDGE_LOOP('', (#3317, #3326, #3333, #3340));
#3342=FACE_BOUND('', #3341, .F.);
#3343=CARTESIAN_POINT('', (260.0, -10.0, 0.0));
#3344=DIRECTION('', (0.0, 1.0, 0.0));
#3345=DIRECTION('', (0.0, 0.0, 1.0));
#3346=AXIS2_PLACEMENT_3D('', #3343, #3344, #3345);
#3347=PLANE('', #3346);
#3348=ADVANCED_FACE('', (#3342), #3347, .T.);
#3349=SHAPE_ASPECT('featured shape: face 75', $, #38, .T.);
#3350=PROPERTY_DEFINITION('', $, #3349);
#3351=FACE_SHAPE_REPRESENTATION('', (#3348), #29);
#3352=PROPERTY_DEFINITION_REPRESENTATION(#3350, #3351);
#3353=ORIENTED_EDGE('', *, *, #931, .F.);
#3354=DIRECTION('', (0.0, 1.0, 0.0));
#3355=VECTOR('', #3354, 10.0);
#3356=LINE('', #3309, #3355);
#3357=EDGE_CURVE('', #3310, #925, #3356, .T.);
#3358=ORIENTED_EDGE('', *, *, #3357, .F.);
#3359=ORIENTED_EDGE('', *, *, #3316, .T.);

```

```

#3360=DIRECTION('', (0.0,1.0,0.0));
#3361=VECTOR('', #3360,10.0);
#3362=LINE('', #3311, #3361);
#3363=EDGE_CURVE('', #3312, #927, #3362, .T.);
#3364=ORIENTED_EDGE('', *, *, #3363, .T.);
#3365=EDGE_LOOP('', (#3353, #3358, #3359, #3364));
#3366=FACE_BOUND('', #3365, .F.);
#3367=CARTESIAN_POINT('', (110.0, -10.0, 85.0));
#3368=DIRECTION('', (0.0,0.0,-1.0));
#3369=DIRECTION('', (-1.0,0.0,0.0));
#3370=AXIS2_PLACEMENT_3D('', #3367, #3368, #3369);
#3371=PLANE('', #3370);
#3372=ADVANCED_FACE('', (#3366), #3371, .F.);
#3373=SHAPE_ASPECT('featured shape: face 76', $, #38, .T.);
#3374=PROPERTY_DEFINITION('', $, #3373);
#3375=FACE_SHAPE_REPRESENTATION('', (#3372), #29);
#3376=PROPERTY_DEFINITION_REPRESENTATION(#3374, #3375);
#3377=ORIENTED_EDGE('', *, *, #940, .F.);
#3378=ORIENTED_EDGE('', *, *, #3363, .F.);
#3379=ORIENTED_EDGE('', *, *, #3339, .T.);
#3380=DIRECTION('', (0.0,1.0,0.0));
#3381=VECTOR('', #3380,10.0);
#3382=LINE('', #3327, #3381);
#3383=EDGE_CURVE('', #3328, #934, #3382, .T.);
#3384=ORIENTED_EDGE('', *, *, #3383, .T.);
#3385=EDGE_LOOP('', (#3377, #3378, #3379, #3384));
#3386=FACE_BOUND('', #3385, .F.);
#3387=CARTESIAN_POINT('', (85.0, -10.0, 90.0));
#3388=DIRECTION('', (0.0,1.0,0.0));
#3389=DIRECTION('', (0.0,0.0,1.0));
#3390=AXIS2_PLACEMENT_3D('', #3387, #3388, #3389);
#3391=CYLINDRICAL_SURFACE('', #3390, 5.0);
#3392=ADVANCED_FACE('', (#3386), #3391, .F.);
#3393=SHAPE_ASPECT('featured shape: face 77', $, #38, .T.);
#3394=PROPERTY_DEFINITION('', $, #3393);
#3395=FACE_SHAPE_REPRESENTATION('', (#3392), #29);
#3396=PROPERTY_DEFINITION_REPRESENTATION(#3394, #3395);
#3397=ORIENTED_EDGE('', *, *, #947, .F.);
#3398=ORIENTED_EDGE('', *, *, #3383, .F.);
#3399=ORIENTED_EDGE('', *, *, #3332, .T.);
#3400=DIRECTION('', (0.0,1.0,0.0));
#3401=VECTOR('', #3400,10.0);
#3402=LINE('', #3318, #3401);
#3403=EDGE_CURVE('', #3319, #943, #3402, .T.);
#3404=ORIENTED_EDGE('', *, *, #3403, .T.);
#3405=EDGE_LOOP('', (#3397, #3398, #3399, #3404));
#3406=FACE_BOUND('', #3405, .F.);
#3407=CARTESIAN_POINT('', (85.0, -10.0, 95.0));
#3408=DIRECTION('', (0.0,0.0,1.0));
#3409=DIRECTION('', (1.0,0.0,0.0));
#3410=AXIS2_PLACEMENT_3D('', #3407, #3408, #3409);
#3411=PLANE('', #3410);
#3412=ADVANCED_FACE('', (#3406), #3411, .F.);

```



```

#3413=SHAPE_ASPECT('featured shape: face 78', $, #38, .T.);
#3414=PROPERTY_DEFINITION('', $, #3413);
#3415=FACE_SHAPE_REPRESENTATION('', (#3412), #29);
#3416=PROPERTY_DEFINITION_REPRESENTATION(#3414, #3415);
#3417=ORIENTED_EDGE('', *, *, #954, .F.);
#3418=ORIENTED_EDGE('', *, *, #3403, .F.);
#3419=ORIENTED_EDGE('', *, *, #3325, .T.);
#3420=ORIENTED_EDGE('', *, *, #3357, .T.);
#3421=EDGE_LOOP('', (#3417, #3418, #3419, #3420));
#3422=FACE_BOUND('', #3421, .F.);
#3423=CARTESIAN_POINT('', (110.0, -10.0, 90.0));
#3424=DIRECTION('', (0.0, 1.0, 0.0));
#3425=DIRECTION('', (0.0, 0.0, 1.0));
#3426=AXIS2_PLACEMENT_3D('', #3423, #3424, #3425);
#3427=CYLINDRICAL_SURFACE('', #3426, 5.0);
#3428=ADVANCED_FACE('', (#3422), #3427, .F.);
#3429=SHAPE_ASPECT('featured shape: face 79', $, #38, .T.);
#3430=PROPERTY_DEFINITION('', $, #3429);
#3431=FACE_SHAPE_REPRESENTATION('', (#3428), #29);
#3432=PROPERTY_DEFINITION_REPRESENTATION(#3430, #3431);
#3433=CARTESIAN_POINT('', (192.568849999999998, -19.050000000000001, 120.0));
#3434=VERTEX_POINT('', #3433);
#3435=CARTESIAN_POINT('', (187.431150000000002, -19.050000000000001, 120.0));
#3436=VERTEX_POINT('', #3435);
#3437=CARTESIAN_POINT('', (190.0, -19.049999999999969, 120.0));
#3438=DIRECTION('', (0.0, -1.0, 0.0));
#3439=DIRECTION('', (1.0, 0.0, 0.0));
#3440=AXIS2_PLACEMENT_3D('', #3437, #3438, #3439);
#3441=CIRCLE('', #3440, 2.568850000000000);
#3442=EDGE_CURVE('', #3434, #3436, #3441, .T.);
#3443=ORIENTED_EDGE('', *, *, #3442, .T.);
#3444=CARTESIAN_POINT('', (190.0, -19.049999999999969, 120.0));
#3445=DIRECTION('', (0.0, -1.0, 0.0));
#3446=DIRECTION('', (-1.0, 0.0, 0.0));
#3447=AXIS2_PLACEMENT_3D('', #3444, #3445, #3446);
#3448=CIRCLE('', #3447, 2.568850000000000);
#3449=EDGE_CURVE('', #3436, #3434, #3448, .T.);
#3450=ORIENTED_EDGE('', *, *, #3449, .T.);
#3451=EDGE_LOOP('', (#3443, #3450));
#3452=FACE_BOUND('', #3451, .F.);
#3453=CARTESIAN_POINT('', (190.0, -19.050000000000001, 120.0));
#3454=DIRECTION('', (0.0, -1.0, 0.0));
#3455=DIRECTION('', (1.0, 0.0, 0.0));
#3456=AXIS2_PLACEMENT_3D('', #3453, #3454, #3455);
#3457=PLANE('', #3456);
#3458=ADVANCED_FACE('', (#3452), #3457, .F.);
#3459=SHAPE_ASPECT('featured shape: face 80', $, #38, .T.);
#3460=PROPERTY_DEFINITION('', $, #3459);
#3461=FACE_SHAPE_REPRESENTATION('', (#3458), #29);
#3462=PROPERTY_DEFINITION_REPRESENTATION(#3460, #3461);
#3463=ORIENTED_EDGE('', *, *, #967, .T.);
#3464=DIRECTION('', (0.0, -1.0, 0.0));
#3465=VECTOR('', #3464, 19.050000000000001);

```

```

#3466=LINE('',#960,#3465);
#3467=EDGE_CURVE('',#961,#3436,#3466,.T.);
#3468=ORIENTED_EDGE('',*,*,#3467,.T.);
#3469=ORIENTED_EDGE('',*,*,#3442,.F.);
#3470=DIRECTION('',(0.0,-1.0,0.0));
#3471=VECTOR('',#3470,19.050000000000001);
#3472=LINE('',#958,#3471);
#3473=EDGE_CURVE('',#959,#3434,#3472,.T.);
#3474=ORIENTED_EDGE('',*,*,#3473,.F.);
#3475=EDGE_LOOP('',(#3463,#3468,#3469,#3474));
#3476=FACE_BOUND('',#3475,.F.);
#3477=CARTESIAN_POINT('',(190.0,0.0,120.0));
#3478=DIRECTION('',(0.0,-1.0,0.0));
#3479=DIRECTION('',(1.0,0.0,0.0));
#3480=AXIS2_PLACEMENT_3D('',#3477,#3478,#3479);
#3481=CYLINDRICAL_SURFACE('',#3480,2.568850000000000);
#3482=ADVANCED_FACE('',(#3476),#3481,.F.);
#3483=SHAPE_ASPECT('featured shape: face 81',$,#38,.T.);
#3484=PROPERTY_DEFINITION('',$,#3483);
#3485=FACE_SHAPE_REPRESENTATION('',(#3482),#29);
#3486=PROPERTY_DEFINITION_REPRESENTATION(#3484,#3485);
#3487=ORIENTED_EDGE('',*,*,#974,.T.);
#3488=ORIENTED_EDGE('',*,*,#3473,.T.);
#3489=ORIENTED_EDGE('',*,*,#3449,.F.);
#3490=ORIENTED_EDGE('',*,*,#3467,.F.);
#3491=EDGE_LOOP('',(#3487,#3488,#3489,#3490));
#3492=FACE_BOUND('',#3491,.F.);
#3493=CARTESIAN_POINT('',(190.0,0.0,120.0));
#3494=DIRECTION('',(0.0,-1.0,0.0));
#3495=DIRECTION('',(1.0,0.0,0.0));
#3496=AXIS2_PLACEMENT_3D('',#3493,#3494,#3495);
#3497=CYLINDRICAL_SURFACE('',#3496,2.568850000000000);
#3498=ADVANCED_FACE('',(#3492),#3497,.F.);
#3499=SHAPE_ASPECT('featured shape: face 82',$,#38,.T.);
#3500=PROPERTY_DEFINITION('',$,#3499);
#3501=FACE_SHAPE_REPRESENTATION('',(#3498),#29);
#3502=PROPERTY_DEFINITION_REPRESENTATION(#3500,#3501);
#3503=CARTESIAN_POINT('',(192.568849999999998,-19.050000000000001,60.0));
#3504=VERTEX_POINT('',#3503);
#3505=CARTESIAN_POINT('',(187.431150000000002,-19.050000000000001,60.0));
#3506=VERTEX_POINT('',#3505);
#3507=CARTESIAN_POINT('',(190.0,-19.049999999999969,60.0));
#3508=DIRECTION('',(0.0,-1.0,0.0));
#3509=DIRECTION('',(1.0,0.0,0.0));
#3510=AXIS2_PLACEMENT_3D('',#3507,#3508,#3509);
#3511=CIRCLE('',#3510,2.568850000000000);
#3512=EDGE_CURVE('',#3504,#3506,#3511,.T.);
#3513=ORIENTED_EDGE('',*,*,#3512,.T.);
#3514=CARTESIAN_POINT('',(190.0,-19.049999999999969,60.0));
#3515=DIRECTION('',(0.0,-1.0,0.0));
#3516=DIRECTION('',(-1.0,0.0,0.0));
#3517=AXIS2_PLACEMENT_3D('',#3514,#3515,#3516);
#3518=CIRCLE('',#3517,2.568850000000000);

```

```

#3519=EDGE_CURVE('',#3506,#3504,#3518,.T.);
#3520=ORIENTED_EDGE('',*,*,#3519,.T.);
#3521=EDGE_LOOP('',(#3513,#3520));
#3522=FACE_BOUND('',#3521,.F.);
#3523=CARTESIAN_POINT('',(190.0,-19.050000000000001,60.0));
#3524=DIRECTION('',(0.0,-1.0,0.0));
#3525=DIRECTION('',(1.0,0.0,0.0));
#3526=AXIS2_PLACEMENT_3D('',#3523,#3524,#3525);
#3527=PLANE('',#3526);
#3528=ADVANCED_FACE('',(#3522),#3527,.F.);
#3529=SHAPE_ASPECT('featured shape: face 83',$,#38,.T.);
#3530=PROPERTY_DEFINITION('',$,#3529);
#3531=FACE_SHAPE_REPRESENTATION('',(#3528),#29);
#3532=PROPERTY_DEFINITION_REPRESENTATION(#3530,#3531);
#3533=ORIENTED_EDGE('',*,*,#987,.T.);
#3534=DIRECTION('',(0.0,-1.0,0.0));
#3535=VECTOR('',#3534,19.050000000000001);
#3536=LINE('',#980,#3535);
#3537=EDGE_CURVE('',#981,#3506,#3536,.T.);
#3538=ORIENTED_EDGE('',*,*,#3537,.T.);
#3539=ORIENTED_EDGE('',*,*,#3512,.F.);
#3540=DIRECTION('',(0.0,-1.0,0.0));
#3541=VECTOR('',#3540,19.050000000000001);
#3542=LINE('',#978,#3541);
#3543=EDGE_CURVE('',#979,#3504,#3542,.T.);
#3544=ORIENTED_EDGE('',*,*,#3543,.F.);
#3545=EDGE_LOOP('',(#3533,#3538,#3539,#3544));
#3546=FACE_BOUND('',#3545,.F.);
#3547=CARTESIAN_POINT('',(190.0,0.0,60.0));
#3548=DIRECTION('',(0.0,-1.0,0.0));
#3549=DIRECTION('',(1.0,0.0,0.0));
#3550=AXIS2_PLACEMENT_3D('',#3547,#3548,#3549);
#3551=CYLINDRICAL_SURFACE('',#3550,2.568850000000000);
#3552=ADVANCED_FACE('',(#3546),#3551,.F.);
#3553=SHAPE_ASPECT('featured shape: face 84',$,#38,.T.);
#3554=PROPERTY_DEFINITION('',$,#3553);
#3555=FACE_SHAPE_REPRESENTATION('',(#3552),#29);
#3556=PROPERTY_DEFINITION_REPRESENTATION(#3554,#3555);
#3557=ORIENTED_EDGE('',*,*,#994,.T.);
#3558=ORIENTED_EDGE('',*,*,#3543,.T.);
#3559=ORIENTED_EDGE('',*,*,#3519,.F.);
#3560=ORIENTED_EDGE('',*,*,#3537,.F.);
#3561=EDGE_LOOP('',(#3557,#3558,#3559,#3560));
#3562=FACE_BOUND('',#3561,.F.);
#3563=CARTESIAN_POINT('',(190.0,0.0,60.0));
#3564=DIRECTION('',(0.0,-1.0,0.0));
#3565=DIRECTION('',(1.0,0.0,0.0));
#3566=AXIS2_PLACEMENT_3D('',#3563,#3564,#3565);
#3567=CYLINDRICAL_SURFACE('',#3566,2.568850000000000);
#3568=ADVANCED_FACE('',(#3562),#3567,.F.);
#3569=SHAPE_ASPECT('featured shape: face 85',$,#38,.T.);
#3570=PROPERTY_DEFINITION('',$,#3569);
#3571=FACE_SHAPE_REPRESENTATION('',(#3568),#29);

```

```

#3572=PROPERTY_DEFINITION_REPRESENTATION(#3570,#3571);
#3573=CARTESIAN_POINT('',(132.568849999999998,-19.050000000000001,120.0));
#3574=VERTEX_POINT('',#3573);
#3575=CARTESIAN_POINT('',(127.431150000000002,-19.050000000000001,120.0));
#3576=VERTEX_POINT('',#3575);
#3577=CARTESIAN_POINT('',(130.0,-19.049999999999969,120.0));
#3578=DIRECTION('',(0.0,-1.0,0.0));
#3579=DIRECTION('',(1.0,0.0,0.0));
#3580=AXIS2_PLACEMENT_3D('',#3577,#3578,#3579);
#3581=CIRCLE('',#3580,2.568850000000000);
#3582=EDGE_CURVE('',#3574,#3576,#3581,.T.);
#3583=ORIENTED_EDGE('',*,*,#3582,.T.);
#3584=CARTESIAN_POINT('',(130.0,-19.049999999999969,120.0));
#3585=DIRECTION('',(0.0,-1.0,0.0));
#3586=DIRECTION('',(-1.0,0.0,0.0));
#3587=AXIS2_PLACEMENT_3D('',#3584,#3585,#3586);
#3588=CIRCLE('',#3587,2.568850000000000);
#3589=EDGE_CURVE('',#3576,#3574,#3588,.T.);
#3590=ORIENTED_EDGE('',*,*,#3589,.T.);
#3591=EDGE_LOOP('',(#3583,#3590));
#3592=FACE_BOUND('',#3591,.F.);
#3593=CARTESIAN_POINT('',(130.0,-19.050000000000001,120.0));
#3594=DIRECTION('',(0.0,-1.0,0.0));
#3595=DIRECTION('',(1.0,0.0,0.0));
#3596=AXIS2_PLACEMENT_3D('',#3593,#3594,#3595);
#3597=PLANE('',#3596);
#3598=ADVANCED_FACE('',(#3592),#3597,.F.);
#3599=SHAPE_ASPECT('featured shape: face 86',$,#38,.T.);
#3600=PROPERTY_DEFINITION('',$,#3599);
#3601=FACE_SHAPE_REPRESENTATION('',(#3598),#29);
#3602=PROPERTY_DEFINITION_REPRESENTATION(#3600,#3601);
#3603=ORIENTED_EDGE('',*,*,#1007,.T.);
#3604=DIRECTION('',(0.0,-1.0,0.0));
#3605=VECTOR('',#3604,19.050000000000001);
#3606=LINE('',#1000,#3605);
#3607=EDGE_CURVE('',#1001,#3576,#3606,.T.);
#3608=ORIENTED_EDGE('',*,*,#3607,.T.);
#3609=ORIENTED_EDGE('',*,*,#3582,.F.);
#3610=DIRECTION('',(0.0,-1.0,0.0));
#3611=VECTOR('',#3610,19.050000000000001);
#3612=LINE('',#998,#3611);
#3613=EDGE_CURVE('',#999,#3574,#3612,.T.);
#3614=ORIENTED_EDGE('',*,*,#3613,.F.);
#3615=EDGE_LOOP('',(#3603,#3608,#3609,#3614));
#3616=FACE_BOUND('',#3615,.F.);
#3617=CARTESIAN_POINT('',(130.0,0.0,120.0));
#3618=DIRECTION('',(0.0,-1.0,0.0));
#3619=DIRECTION('',(1.0,0.0,0.0));
#3620=AXIS2_PLACEMENT_3D('',#3617,#3618,#3619);
#3621=CYLINDRICAL_SURFACE('',#3620,2.568850000000000);
#3622=ADVANCED_FACE('',(#3616),#3621,.F.);
#3623=SHAPE_ASPECT('featured shape: face 87',$,#38,.T.);
#3624=PROPERTY_DEFINITION('',$,#3623);

```

```

#3625=FACE_SHAPE_REPRESENTATION('', (#3622), #29);
#3626=PROPERTY_DEFINITION_REPRESENTATION(#3624, #3625);
#3627=ORIENTED_EDGE('', *, *, #1014, .T.);
#3628=ORIENTED_EDGE('', *, *, #3613, .T.);
#3629=ORIENTED_EDGE('', *, *, #3589, .F.);
#3630=ORIENTED_EDGE('', *, *, #3607, .F.);
#3631=EDGE_LOOP('', (#3627, #3628, #3629, #3630));
#3632=FACE_BOUND('', #3631, .F.);
#3633=CARTESIAN_POINT('', (130.0, 0.0, 120.0));
#3634=DIRECTION('', (0.0, -1.0, 0.0));
#3635=DIRECTION('', (1.0, 0.0, 0.0));
#3636=AXIS2_PLACEMENT_3D('', #3633, #3634, #3635);
#3637=CYLINDRICAL_SURFACE('', #3636, 2.5688500000000000);
#3638=ADVANCED_FACE('', (#3632), #3637, .F.);
#3639=SHAPE_ASPECT('featured shape: face 88', $, #38, .T.);
#3640=PROPERTY_DEFINITION('', $, #3639);
#3641=FACE_SHAPE_REPRESENTATION('', (#3638), #29);
#3642=PROPERTY_DEFINITION_REPRESENTATION(#3640, #3641);
#3643=CARTESIAN_POINT('', (132.568849999999998, -19.050000000000001, 60.0));
#3644=VERTEX_POINT('', #3643);
#3645=CARTESIAN_POINT('', (127.431150000000002, -19.050000000000001, 60.0));
#3646=VERTEX_POINT('', #3645);
#3647=CARTESIAN_POINT('', (130.0, -19.049999999999969, 60.0));
#3648=DIRECTION('', (0.0, -1.0, 0.0));
#3649=DIRECTION('', (1.0, 0.0, 0.0));
#3650=AXIS2_PLACEMENT_3D('', #3647, #3648, #3649);
#3651=CIRCLE('', #3650, 2.5688500000000000);
#3652=EDGE_CURVE('', #3644, #3646, #3651, .T.);
#3653=ORIENTED_EDGE('', *, *, #3652, .T.);
#3654=CARTESIAN_POINT('', (130.0, -19.049999999999969, 60.0));
#3655=DIRECTION('', (0.0, -1.0, 0.0));
#3656=DIRECTION('', (-1.0, 0.0, 0.0));
#3657=AXIS2_PLACEMENT_3D('', #3654, #3655, #3656);
#3658=CIRCLE('', #3657, 2.5688500000000000);
#3659=EDGE_CURVE('', #3646, #3644, #3658, .T.);
#3660=ORIENTED_EDGE('', *, *, #3659, .T.);
#3661=EDGE_LOOP('', (#3653, #3660));
#3662=FACE_BOUND('', #3661, .F.);
#3663=CARTESIAN_POINT('', (130.0, -19.050000000000001, 60.0));
#3664=DIRECTION('', (0.0, -1.0, 0.0));
#3665=DIRECTION('', (1.0, 0.0, 0.0));
#3666=AXIS2_PLACEMENT_3D('', #3663, #3664, #3665);
#3667=PLANE('', #3666);
#3668=ADVANCED_FACE('', (#3662), #3667, .F.);
#3669=SHAPE_ASPECT('featured shape: face 89', $, #38, .T.);
#3670=PROPERTY_DEFINITION('', $, #3669);
#3671=FACE_SHAPE_REPRESENTATION('', (#3668), #29);
#3672=PROPERTY_DEFINITION_REPRESENTATION(#3670, #3671);
#3673=ORIENTED_EDGE('', *, *, #1027, .T.);
#3674=DIRECTION('', (0.0, -1.0, 0.0));
#3675=VECTOR('', #3674, 19.050000000000001);
#3676=LINE('', #1020, #3675);
#3677=EDGE_CURVE('', #1021, #3646, #3676, .T.);

```

```

#3678=ORIENTED_EDGE('',*,*,#3677,.T.);
#3679=ORIENTED_EDGE('',*,*,#3652,.F.);
#3680=DIRECTION('',(0.0,-1.0,0.0));
#3681=VECTOR('',#3680,19.050000000000001);
#3682=LINE('',#1018,#3681);
#3683=EDGE_CURVE('',#1019,#3644,#3682,.T.);
#3684=ORIENTED_EDGE('',*,*,#3683,.F.);
#3685=EDGE_LOOP('',(#3673,#3678,#3679,#3684));
#3686=FACE_BOUND('',#3685,.F.);
#3687=CARTESIAN_POINT('',(130.0,0.0,60.0));
#3688=DIRECTION('',(0.0,-1.0,0.0));
#3689=DIRECTION('',(1.0,0.0,0.0));
#3690=AXIS2_PLACEMENT_3D('',#3687,#3688,#3689);
#3691=CYLINDRICAL_SURFACE('',#3690,2.568850000000000);
#3692=ADVANCED_FACE('',(#3686),#3691,.F.);
#3693=SHAPE_ASPECT('featured shape: face 90',$,#38,.T.);
#3694=PROPERTY_DEFINITION('',$,#3693);
#3695=FACE_SHAPE_REPRESENTATION('',(#3692),#29);
#3696=PROPERTY_DEFINITION_REPRESENTATION(#3694,#3695);
#3697=ORIENTED_EDGE('',*,*,#1034,.T.);
#3698=ORIENTED_EDGE('',*,*,#3683,.T.);
#3699=ORIENTED_EDGE('',*,*,#3659,.F.);
#3700=ORIENTED_EDGE('',*,*,#3677,.F.);
#3701=EDGE_LOOP('',(#3697,#3698,#3699,#3700));
#3702=FACE_BOUND('',#3701,.F.);
#3703=CARTESIAN_POINT('',(130.0,0.0,60.0));
#3704=DIRECTION('',(0.0,-1.0,0.0));
#3705=DIRECTION('',(1.0,0.0,0.0));
#3706=AXIS2_PLACEMENT_3D('',#3703,#3704,#3705);
#3707=CYLINDRICAL_SURFACE('',#3706,2.568850000000000);
#3708=ADVANCED_FACE('',(#3702),#3707,.F.);
#3709=SHAPE_ASPECT('featured shape: face 91',$,#38,.T.);
#3710=PROPERTY_DEFINITION('',$,#3709);
#3711=FACE_SHAPE_REPRESENTATION('',(#3708),#29);
#3712=PROPERTY_DEFINITION_REPRESENTATION(#3710,#3711);
#3713=ORIENTED_EDGE('',*,*,#1210,.F.);
#3714=ORIENTED_EDGE('',*,*,#735,.T.);
#3715=ORIENTED_EDGE('',*,*,#53,.T.);
#3716=ORIENTED_EDGE('',*,*,#1443,.F.);
#3717=EDGE_LOOP('',(#3713,#3714,#3715,#3716));
#3718=FACE_BOUND('',#3717,.F.);
#3719=CARTESIAN_POINT('',(225.0,0.0,5.0));
#3720=DIRECTION('',(0.0,1.0,0.0));
#3721=DIRECTION('',(1.0,0.0,0.0));
#3722=AXIS2_PLACEMENT_3D('',#3719,#3720,#3721);
#3723=CYLINDRICAL_SURFACE('',#3722,5.0);
#3724=ADVANCED_FACE('',(#3718),#3723,.T.);
#3725=SHAPE_ASPECT('featured shape: face 92',$,#38,.T.);
#3726=PROPERTY_DEFINITION('',$,#3725);
#3727=FACE_SHAPE_REPRESENTATION('',(#3724),#29);
#3728=PROPERTY_DEFINITION_REPRESENTATION(#3726,#3727);
#3729=ORIENTED_EDGE('',*,*,#1250,.F.);
#3730=ORIENTED_EDGE('',*,*,#1387,.T.);

```

```

#3731=ORIENTED_EDGE('',*,*,#107,.T.);
#3732=ORIENTED_EDGE('',*,*,#779,.F.);
#3733=EDGE_LOOP('',(#3729,#3730,#3731,#3732));
#3734=FACE_BOUND('',#3733,.F.);
#3735=CARTESIAN_POINT('',(225.0,-40.0,175.0));
#3736=DIRECTION('',(0.0,-1.0,0.0));
#3737=DIRECTION('',(1.0,0.0,0.0));
#3738=AXIS2_PLACEMENT_3D('',#3735,#3736,#3737);
#3739=CYLINDRICAL_SURFACE('',#3738,5.0);
#3740=ADVANCED_FACE('',(#3734),#3739,.T.);
#3741=SHAPE_ASPECT('featured shape: face 93',$,#38,.T.);
#3742=PROPERTY_DEFINITION('',$,#3741);
#3743=FACE_SHAPE_REPRESENTATION('',(#3740),#29);
#3744=PROPERTY_DEFINITION_REPRESENTATION(#3742,#3743);
#3745=ORIENTED_EDGE('',*,*,#121,.T.);
#3746=ORIENTED_EDGE('',*,*,#1379,.T.);
#3747=ORIENTED_EDGE('',*,*,#182,.F.);
#3748=ORIENTED_EDGE('',*,*,#695,.F.);
#3749=EDGE_LOOP('',(#3745,#3746,#3747,#3748));
#3750=FACE_BOUND('',#3749,.F.);
#3751=CARTESIAN_POINT('',(5.0,0.0,175.0));
#3752=DIRECTION('',(0.0,-1.0,0.0));
#3753=DIRECTION('',(0.0,0.0,1.0));
#3754=AXIS2_PLACEMENT_3D('',#3751,#3752,#3753);
#3755=CYLINDRICAL_SURFACE('',#3754,5.0);
#3756=ADVANCED_FACE('',(#3750),#3755,.T.);
#3757=SHAPE_ASPECT('featured shape: face 94',$,#38,.T.);
#3758=PROPERTY_DEFINITION('',$,#3757);
#3759=FACE_SHAPE_REPRESENTATION('',(#3756),#29);
#3760=PROPERTY_DEFINITION_REPRESENTATION(#3758,#3759);
#3761=ORIENTED_EDGE('',*,*,#65,.F.);
#3762=ORIENTED_EDGE('',*,*,#725,.T.);
#3763=ORIENTED_EDGE('',*,*,#196,.T.);
#3764=ORIENTED_EDGE('',*,*,#1371,.F.);
#3765=EDGE_LOOP('',(#3761,#3762,#3763,#3764));
#3766=FACE_BOUND('',#3765,.F.);
#3767=CARTESIAN_POINT('',(5.0,0.0,5.0));
#3768=DIRECTION('',(0.0,1.0,0.0));
#3769=DIRECTION('',(0.0,0.0,-1.0));
#3770=AXIS2_PLACEMENT_3D('',#3767,#3768,#3769);
#3771=CYLINDRICAL_SURFACE('',#3770,5.0);
#3772=ADVANCED_FACE('',(#3766),#3771,.T.);
#3773=SHAPE_ASPECT('featured shape: face 95',$,#38,.T.);
#3774=PROPERTY_DEFINITION('',$,#3773);
#3775=FACE_SHAPE_REPRESENTATION('',(#3772),#29);
#3776=PROPERTY_DEFINITION_REPRESENTATION(#3774,#3775);
#3777=ORIENTED_EDGE('',*,*,#619,.T.);
#3778=ORIENTED_EDGE('',*,*,#1286,.T.);
#3779=ORIENTED_EDGE('',*,*,#1864,.F.);
#3780=ORIENTED_EDGE('',*,*,#1421,.F.);
#3781=EDGE_LOOP('',(#3777,#3778,#3779,#3780));
#3782=FACE_BOUND('',#3781,.F.);
#3783=CARTESIAN_POINT('',(255.0,-40.0,55.0));

```

ISO 10303-223:2008 (E)

```

#3784=DIRECTION('', (0.0,1.0,0.0));
#3785=DIRECTION('', (1.0,0.0,0.0));
#3786=AXIS2_PLACEMENT_3D('', #3783, #3784, #3785);
#3787=CYLINDRICAL_SURFACE('', #3786, 5.0);
#3788=ADVANCED_FACE('', (#3782), #3787, .T.);
#3789=SHAPE_ASPECT('featured shape: face 96', $, #38, .T.);
#3790=PROPERTY_DEFINITION('', $, #3789);
#3791=FACE_SHAPE_REPRESENTATION('', (#3788), #29);
#3792=PROPERTY_DEFINITION_REPRESENTATION(#3790, #3791);
#3793=ORIENTED_EDGE('', *, *, #633, .F.);
#3794=ORIENTED_EDGE('', *, *, #1411, .T.);
#3795=ORIENTED_EDGE('', *, *, #1824, .T.);
#3796=ORIENTED_EDGE('', *, *, #1296, .F.);
#3797=EDGE_LOOP('', (#3793, #3794, #3795, #3796));
#3798=FACE_BOUND('', #3797, .F.);
#3799=CARTESIAN_POINT('', (255.0, -40.0, 125.0));
#3800=DIRECTION('', (0.0, -1.0, 0.0));
#3801=DIRECTION('', (1.0, 0.0, 0.0));
#3802=AXIS2_PLACEMENT_3D('', #3799, #3800, #3801);
#3803=CYLINDRICAL_SURFACE('', #3802, 5.0);
#3804=ADVANCED_FACE('', (#3798), #3803, .T.);
#3805=SHAPE_ASPECT('featured shape: face 97', $, #38, .T.);
#3806=PROPERTY_DEFINITION('', $, #3805);
#3807=FACE_SHAPE_REPRESENTATION('', (#3804), #29);
#3808=PROPERTY_DEFINITION_REPRESENTATION(#3806, #3807);
#3809=ORIENTED_EDGE('', *, *, #2148, .F.);
#3810=ORIENTED_EDGE('', *, *, #478, .T.);
#3811=ORIENTED_EDGE('', *, *, #2210, .T.);
#3812=ORIENTED_EDGE('', *, *, #1641, .F.);
#3813=EDGE_LOOP('', (#3809, #3810, #3811, #3812));
#3814=FACE_BOUND('', #3813, .F.);
#3815=CARTESIAN_POINT('', (223.0, -50.0, 173.0));
#3816=DIRECTION('', (0.0, -1.0, 0.0));
#3817=DIRECTION('', (1.0, 0.0, 0.0));
#3818=AXIS2_PLACEMENT_3D('', #3815, #3816, #3817);
#3819=CYLINDRICAL_SURFACE('', #3818, 5.0);
#3820=ADVANCED_FACE('', (#3814), #3819, .T.);
#3821=SHAPE_ASPECT('featured shape: face 98', $, #38, .T.);
#3822=PROPERTY_DEFINITION('', $, #3821);
#3823=FACE_SHAPE_REPRESENTATION('', (#3820), #29);
#3824=PROPERTY_DEFINITION_REPRESENTATION(#3822, #3823);
#3825=ORIENTED_EDGE('', *, *, #2124, .T.);
#3826=ORIENTED_EDGE('', *, *, #1575, .T.);
#3827=ORIENTED_EDGE('', *, *, #2277, .F.);
#3828=ORIENTED_EDGE('', *, *, #387, .F.);
#3829=EDGE_LOOP('', (#3825, #3826, #3827, #3828));
#3830=FACE_BOUND('', #3829, .F.);
#3831=CARTESIAN_POINT('', (223.0, -50.0, 7.0));
#3832=DIRECTION('', (0.0, 1.0, 0.0));
#3833=DIRECTION('', (1.0, 0.0, 0.0));
#3834=AXIS2_PLACEMENT_3D('', #3831, #3832, #3833);
#3835=CYLINDRICAL_SURFACE('', #3834, 5.0);
#3836=ADVANCED_FACE('', (#3830), #3835, .T.);

```



```

#3837=SHAPE_ASPECT('featured shape: face 99', $, #38, .T.);
#3838=PROPERTY_DEFINITION('', $, #3837);
#3839=FACE_SHAPE_REPRESENTATION('', (#3836), #29);
#3840=PROPERTY_DEFINITION_REPRESENTATION(#3838, #3839);
#3841=ORIENTED_EDGE('', *, *, #2191, .T.);
#3842=ORIENTED_EDGE('', *, *, #1787, .T.);
#3843=ORIENTED_EDGE('', *, *, #2234, .F.);
#3844=ORIENTED_EDGE('', *, *, #569, .F.);
#3845=EDGE_LOOP('', (#3841, #3842, #3843, #3844));
#3846=FACE_BOUND('', #3845, .F.);
#3847=CARTESIAN_POINT('', (7.0, -50.0, 173.0));
#3848=DIRECTION('', (0.0, 1.0, 0.0));
#3849=DIRECTION('', (-1.0, 0.0, 0.0));
#3850=AXIS2_PLACEMENT_3D('', #3847, #3848, #3849);
#3851=CYLINDRICAL_SURFACE('', #3850, 5.0);
#3852=ADVANCED_FACE('', (#3846), #3851, .T.);
#3853=SHAPE_ASPECT('featured shape: face 100', $, #38, .T.);
#3854=PROPERTY_DEFINITION('', $, #3853);
#3855=FACE_SHAPE_REPRESENTATION('', (#3852), #29);
#3856=PROPERTY_DEFINITION_REPRESENTATION(#3854, #3855);
#3857=ORIENTED_EDGE('', *, *, #2167, .F.);
#3858=ORIENTED_EDGE('', *, *, #291, .T.);
#3859=ORIENTED_EDGE('', *, *, #2253, .T.);
#3860=ORIENTED_EDGE('', *, *, #1509, .F.);
#3861=EDGE_LOOP('', (#3857, #3858, #3859, #3860));
#3862=FACE_BOUND('', #3861, .F.);
#3863=CARTESIAN_POINT('', (7.0, -50.0, 7.0));
#3864=DIRECTION('', (0.0, -1.0, 0.0));
#3865=DIRECTION('', (-1.0, 0.0, 0.0));
#3866=AXIS2_PLACEMENT_3D('', #3863, #3864, #3865);
#3867=CYLINDRICAL_SURFACE('', #3866, 5.0);
#3868=ADVANCED_FACE('', (#3862), #3867, .T.);
#3869=SHAPE_ASPECT('featured shape: face 101', $, #38, .T.);
#3870=PROPERTY_DEFINITION('', $, #3869);
#3871=FACE_SHAPE_REPRESENTATION('', (#3868), #29);
#3872=PROPERTY_DEFINITION_REPRESENTATION(#3870, #3871);
#3873=ORIENTED_EDGE('', *, *, #1045, .F.);
#3874=CARTESIAN_POINT('', (99.999999999999986, -15.609999999999999, 10.0));
#3875=VERTEX_POINT('', #3874);
#3876=DIRECTION('', (0.0, 1.0, 0.0));
#3877=VECTOR('', #3876, 15.610000000000003);
#3878=LINE('', #3874, #3877);
#3879=EDGE_CURVE('', #3875, #1039, #3878, .T.);
#3880=ORIENTED_EDGE('', *, *, #3879, .F.);
#3881=CARTESIAN_POINT('', (91.065103839083633, -12.632558739516522, 10.0));
#3882=VERTEX_POINT('', #3881);
#3883=DIRECTION('', (-0.948710608132913, 0.316145823973808, 0.0));
#3884=VECTOR('', #3883, 9.417936391056951);
#3885=LINE('', #3874, #3884);
#3886=EDGE_CURVE('', #3875, #3882, #3885, .T.);
#3887=ORIENTED_EDGE('', *, *, #3886, .T.);
#3888=CARTESIAN_POINT('', (68.934896160916992, -12.632558739516311, 10.0));
#3889=VERTEX_POINT('', #3888);

```

```

#3890=CARTESIAN_POINT('', (80.0, -45.837430024168363, 10.0));
#3891=DIRECTION('', (0.0, 0.0, 1.0));
#3892=DIRECTION('', (0.316145823973818, 0.948710608132910, 0.0));
#3893=AXIS2_PLACEMENT_3D('', #3890, #3891, #3892);
#3894=CIRCLE('', #3893, 35.0000000000004434);
#3895=EDGE_CURVE('', #3882, #3889, #3894, .T.);
#3896=ORIENTED_EDGE('', *, *, #3895, .T.);
#3897=CARTESIAN_POINT('', (60.0, -15.610000000000003, 10.0));
#3898=VERTEX_POINT('', #3897);
#3899=DIRECTION('', (-0.948710608132913, -0.316145823973808, 0.0));
#3900=VECTOR('', #3899, 9.417936391056960);
#3901=LINE('', #3888, #3900);
#3902=EDGE_CURVE('', #3889, #3898, #3901, .T.);
#3903=ORIENTED_EDGE('', *, *, #3902, .T.);
#3904=DIRECTION('', (4.551843E-16, -1.0, 0.0));
#3905=VECTOR('', #3904, 15.610000000000001);
#3906=LINE('', #1040, #3905);
#3907=EDGE_CURVE('', #1041, #3898, #3906, .T.);
#3908=ORIENTED_EDGE('', *, *, #3907, .F.);
#3909=EDGE_LOOP('', (#3873, #3880, #3887, #3896, #3903, #3908));
#3910=FACE_BOUND('', #3909, .F.);
#3911=CARTESIAN_POINT('', (50.0, 0.0, 10.0));
#3912=DIRECTION('', (0.0, 0.0, 1.0));
#3913=DIRECTION('', (1.0, 0.0, 0.0));
#3914=AXIS2_PLACEMENT_3D('', #3911, #3912, #3913);
#3915=PLANE('', #3914);
#3916=ADVANCED_FACE('', (#3910), #3915, .T.);
#3917=SHAPE_ASPECT('featured shape: face 102', $, #38, .T.);
#3918=PROPERTY_DEFINITION('', $, #3917);
#3919=FACE_SHAPE_REPRESENTATION('', (#3916), #29);
#3920=PROPERTY_DEFINITION_REPRESENTATION(#3918, #3919);
#3921=ORIENTED_EDGE('', *, *, #1093, .F.);
#3922=CARTESIAN_POINT('', (109.99999999999986, -15.610000000000001, 50.0));
#3923=VERTEX_POINT('', #3922);
#3924=DIRECTION('', (0.0, 1.0, 0.0));
#3925=VECTOR('', #3924, 15.610000000000001);
#3926=LINE('', #3922, #3925);
#3927=EDGE_CURVE('', #3923, #1080, #3926, .T.);
#3928=ORIENTED_EDGE('', *, *, #3927, .F.);
#3929=CARTESIAN_POINT('', (109.99999999999986, -15.609999999999999, 20.0));
#3930=VERTEX_POINT('', #3929);
#3931=DIRECTION('', (0.0, 0.0, -1.0));
#3932=VECTOR('', #3931, 30.0);
#3933=LINE('', #3922, #3932);
#3934=EDGE_CURVE('', #3923, #3930, #3933, .T.);
#3935=ORIENTED_EDGE('', *, *, #3934, .T.);
#3936=DIRECTION('', (0.0, 1.0, 0.0));
#3937=VECTOR('', #3936, 15.609999999999999);
#3938=LINE('', #3929, #3937);
#3939=EDGE_CURVE('', #3930, #1089, #3938, .T.);
#3940=ORIENTED_EDGE('', *, *, #3939, .T.);
#3941=EDGE_LOOP('', (#3921, #3928, #3935, #3940));
#3942=FACE_BOUND('', #3941, .F.);

```

```

#3943=CARTESIAN_POINT('',(109.999999999999986,-15.609999999999999,90.0));
#3944=DIRECTION('',(1.0,0.0,0.0));
#3945=DIRECTION('',(0.0,1.0,0.0));
#3946=AXIS2_PLACEMENT_3D('','#3943,#3944,#3945);
#3947=PLANE('','#3946);
#3948=ADVANCED_FACE('',( #3942), #3947, .F.);
#3949=SHAPE_ASPECT('featured shape: face 103', $, #38, .T.);
#3950=PROPERTY_DEFINITION('',$, #3949);
#3951=FACE_SHAPE_REPRESENTATION('',( #3948), #29);
#3952=PROPERTY_DEFINITION_REPRESENTATION(#3950, #3951);
#3953=CARTESIAN_POINT('',(96.838541760261919,-25.097106081329134,20.0));
#3954=VERTEX_POINT('','#3953);
#3955=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,20.0));
#3956=DIRECTION('',(0.0,0.0,-1.0));
#3957=DIRECTION('',(1.0,-1.776357E-16,0.0));
#3958=AXIS2_PLACEMENT_3D('','#3955,#3956,#3957);
#3959=CIRCLE('','#3958,10.0);
#3960=EDGE_CURVE('','#3930,#3954,#3959, .T.);
#3961=ORIENTED_EDGE('',*,*, #3960, .F.);
#3962=ORIENTED_EDGE('',*,*, #3934, .F.);
#3963=CARTESIAN_POINT('',(96.838541760261919,-25.097106081329134,50.0));
#3964=VERTEX_POINT('','#3963);
#3965=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,50.0));
#3966=DIRECTION('',(0.0,0.0,1.000000000000000));
#3967=DIRECTION('',(-0.316145823973807,-0.948710608132914,0.0));
#3968=AXIS2_PLACEMENT_3D('','#3965,#3966,#3967);
#3969=CIRCLE('','#3968,10.0);
#3970=EDGE_CURVE('','#3964,#3923,#3969, .T.);
#3971=ORIENTED_EDGE('',*,*, #3970, .F.);
#3972=DIRECTION('',(0.0,0.0,-1.0));
#3973=VECTOR('','#3972,30.0);
#3974=LINE('','#3963,#3973);
#3975=EDGE_CURVE('','#3964,#3954,#3974, .T.);
#3976=ORIENTED_EDGE('',*,*, #3975, .T.);
#3977=EDGE_LOOP('',( #3961, #3962, #3971, #3976));
#3978=FACE_BOUND('','#3977, .F.);
#3979=CARTESIAN_POINT('',(99.999999999999986,-15.609999999999999,90.0));
#3980=DIRECTION('',(0.0,0.0,1.0));
#3981=DIRECTION('',(-1.0,0.0,0.0));
#3982=AXIS2_PLACEMENT_3D('','#3979,#3980,#3981);
#3983=CYLINDRICAL_SURFACE('','#3982,10.0);
#3984=ADVANCED_FACE('',( #3978), #3983, .F.);
#3985=SHAPE_ASPECT('featured shape: face 104', $, #38, .T.);
#3986=PROPERTY_DEFINITION('',$, #3985);
#3987=FACE_SHAPE_REPRESENTATION('',( #3984), #29);
#3988=PROPERTY_DEFINITION_REPRESENTATION(#3986, #3987);
#3989=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,20.0));
#3990=VERTEX_POINT('','#3989);
#3991=DIRECTION('',(-0.948710608132913,0.316145823973807,0.0));
#3992=VECTOR('','#3991,9.417936391056967);
#3993=LINE('','#3953,#3992);
#3994=EDGE_CURVE('','#3954,#3990,#3993, .T.);
#3995=ORIENTED_EDGE('',*,*, #3994, .F.);

```

```

#3996=ORIENTED_EDGE('',*,*,#3975,.F.);
#3997=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,50.0));
#3998=VERTEX_POINT('',#3997);
#3999=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4000=VECTOR('',#3999,9.417936391056950);
#4001=LINE('',#3963,#4000);
#4002=EDGE_CURVE('',#3964,#3998,#4001,.T.);
#4003=ORIENTED_EDGE('',*,*,#4002,.T.);
#4004=DIRECTION('',(-4.736952E-16,2.368476E-16,-1.0));
#4005=VECTOR('',#4004,30.0);
#4006=LINE('',#3997,#4005);
#4007=EDGE_CURVE('',#3998,#3990,#4006,.T.);
#4008=ORIENTED_EDGE('',*,*,#4007,.T.);
#4009=EDGE_LOOP('',(#3995,#3996,#4003,#4008));
#4010=FACE_BOUND('',#4009,.F.);
#4011=CARTESIAN_POINT('',(87.903645599345182,-22.119664820845532,90.0));
#4012=DIRECTION('',(-0.316145823973807,-0.948710608132913,0.0));
#4013=DIRECTION('',(0.948710608132913,-0.316145823973807,0.0));
#4014=AXIS2_PLACEMENT_3D('',#4011,#4012,#4013);
#4015=PLANE('',#4014);
#4016=ADVANCED_FACE('',(#4010),#4015,.F.);
#4017=SHAPE_ASPECT('featured shape: face 105',$,#38,.T.);
#4018=PROPERTY_DEFINITION('',$,#4017);
#4019=FACE_SHAPE_REPRESENTATION('',(#4016),#29);
#4020=PROPERTY_DEFINITION_REPRESENTATION(#4018,#4019);
#4021=CARTESIAN_POINT('',(72.096354400654803,-22.119664820845529,20.0));
#4022=VERTEX_POINT('',#4021);
#4023=CARTESIAN_POINT('',(80.0,-45.837430024168363,20.000000000000021));
#4024=DIRECTION('',(0.0,0.0,1.000000000000000));
#4025=DIRECTION('',(0.316145823973807,0.948710608132914,0.0));
#4026=AXIS2_PLACEMENT_3D('',#4023,#4024,#4025);
#4027=CIRCLE('',#4026,25.0);
#4028=EDGE_CURVE('',#3990,#4022,#4027,.T.);
#4029=ORIENTED_EDGE('',*,*,#4028,.F.);
#4030=ORIENTED_EDGE('',*,*,#4007,.F.);
#4031=CARTESIAN_POINT('',(72.096354400654803,-22.119664820845529,50.0));
#4032=VERTEX_POINT('',#4031);
#4033=CARTESIAN_POINT('',(80.0,-45.837430024168363,49.999999999999979));
#4034=DIRECTION('',(0.0,0.0,1.000000000000000));
#4035=DIRECTION('',(0.316145823973808,0.948710608132913,0.0));
#4036=AXIS2_PLACEMENT_3D('',#4033,#4034,#4035);
#4037=CIRCLE('',#4036,24.999999999999993);
#4038=EDGE_CURVE('',#3998,#4032,#4037,.T.);
#4039=ORIENTED_EDGE('',*,*,#4038,.T.);
#4040=DIRECTION('',(0.0,0.0,-1.0));
#4041=VECTOR('',#4040,30.0);
#4042=LINE('',#4031,#4041);
#4043=EDGE_CURVE('',#4032,#4022,#4042,.T.);
#4044=ORIENTED_EDGE('',*,*,#4043,.T.);
#4045=EDGE_LOOP('',(#4029,#4030,#4039,#4044));
#4046=FACE_BOUND('',#4045,.F.);
#4047=CARTESIAN_POINT('',(80.0,-45.837430024168363,90.0));
#4048=DIRECTION('',(0.0,0.0,1.0));

```

```

#4049=DIRECTION('', (-1.0,0.0,0.0));
#4050=AXIS2_PLACEMENT_3D('', #4047, #4048, #4049);
#4051=CYLINDRICAL_SURFACE('', #4050, 25.0);
#4052=ADVANCED_FACE('', (#4046), #4051, .T.);
#4053=SHAPE_ASPECT('featured shape: face 106', $, #38, .T.);
#4054=PROPERTY_DEFINITION('', $, #4053);
#4055=FACE_SHAPE_REPRESENTATION('', (#4052), #29);
#4056=PROPERTY_DEFINITION_REPRESENTATION(#4054, #4055);
#4057=CARTESIAN_POINT('', (63.161458239738074, -25.097106081329137, 20.0));
#4058=VERTEX_POINT('', #4057);
#4059=DIRECTION('', (-0.948710608132914, -0.316145823973807, 0.0));
#4060=VECTOR('', #4059, 9.417936391056950);
#4061=LINE('', #4021, #4060);
#4062=EDGE_CURVE('', #4022, #4058, #4061, .T.);
#4063=ORIENTED_EDGE('', *, *, #4062, .F.);
#4064=ORIENTED_EDGE('', *, *, #4043, .F.);
#4065=CARTESIAN_POINT('', (63.161458239738074, -25.097106081329137, 50.0));
#4066=VERTEX_POINT('', #4065);
#4067=DIRECTION('', (-0.948710608132913, -0.316145823973807, 0.0));
#4068=VECTOR('', #4067, 9.417936391056951);
#4069=LINE('', #4031, #4068);
#4070=EDGE_CURVE('', #4032, #4066, #4069, .T.);
#4071=ORIENTED_EDGE('', *, *, #4070, .T.);
#4072=DIRECTION('', (0.0,0.0,-1.0));
#4073=VECTOR('', #4072, 30.0);
#4074=LINE('', #4065, #4073);
#4075=EDGE_CURVE('', #4066, #4058, #4074, .T.);
#4076=ORIENTED_EDGE('', *, *, #4075, .T.);
#4077=EDGE_LOOP('', (#4063, #4064, #4071, #4076));
#4078=FACE_BOUND('', #4077, .F.);
#4079=CARTESIAN_POINT('', (63.161458239738060, -25.097106081329141, 90.0));
#4080=DIRECTION('', (0.316145823973807, -0.948710608132914, 0.0));
#4081=DIRECTION('', (0.948710608132914, 0.316145823973807, 0.0));
#4082=AXIS2_PLACEMENT_3D('', #4079, #4080, #4081);
#4083=PLANE('', #4082);
#4084=ADVANCED_FACE('', (#4078), #4083, .F.);
#4085=SHAPE_ASPECT('featured shape: face 107', $, #38, .T.);
#4086=PROPERTY_DEFINITION('', $, #4085);
#4087=FACE_SHAPE_REPRESENTATION('', (#4084), #29);
#4088=PROPERTY_DEFINITION_REPRESENTATION(#4086, #4087);
#4089=CARTESIAN_POINT('', (50.0, -15.609999999999998, 20.0));
#4090=VERTEX_POINT('', #4089);
#4091=CARTESIAN_POINT('', (60.0, -15.609999999999992, 20.0));
#4092=DIRECTION('', (0.0,0.0,-1.0));
#4093=DIRECTION('', (0.316145823973808, -0.948710608132913, 0.0));
#4094=AXIS2_PLACEMENT_3D('', #4091, #4092, #4093);
#4095=CIRCLE('', #4094, 10.0);
#4096=EDGE_CURVE('', #4058, #4090, #4095, .T.);
#4097=ORIENTED_EDGE('', *, *, #4096, .F.);
#4098=ORIENTED_EDGE('', *, *, #4075, .F.);
#4099=CARTESIAN_POINT('', (50.0, -15.609999999999998, 50.0));
#4100=VERTEX_POINT('', #4099);
#4101=CARTESIAN_POINT('', (60.0, -15.609999999999992, 50.0));

```

```

#4102=DIRECTION('', (0.0,0.0,1.0));
#4103=DIRECTION('', (-1.0,-5.329071E-16,0.0));
#4104=AXIS2_PLACEMENT_3D('', #4101, #4102, #4103);
#4105=CIRCLE('', #4104, 10.0);
#4106=EDGE_CURVE('', #4100, #4066, #4105, .T.);
#4107=ORIENTED_EDGE('', *, *, #4106, .F.);
#4108=DIRECTION('', (0.0,0.0,-1.0));
#4109=VECTOR('', #4108, 30.0);
#4110=LINE('', #4099, #4109);
#4111=EDGE_CURVE('', #4100, #4090, #4110, .T.);
#4112=ORIENTED_EDGE('', *, *, #4111, .T.);
#4113=EDGE_LOOP('', (#4097, #4098, #4107, #4112));
#4114=FACE_BOUND('', #4113, .F.);
#4115=CARTESIAN_POINT('', (60.0, -15.609999999999992, 90.0));
#4116=DIRECTION('', (0.0,0.0,1.0));
#4117=DIRECTION('', (-1.0,0.0,0.0));
#4118=AXIS2_PLACEMENT_3D('', #4115, #4116, #4117);
#4119=CYLINDRICAL_SURFACE('', #4118, 10.0);
#4120=ADVANCED_FACE('', (#4114), #4119, .F.);
#4121=SHAPE_ASPECT('featured shape: face 108', $, #38, .T.);
#4122=PROPERTY_DEFINITION('', $, #4121);
#4123=FACE_SHAPE_REPRESENTATION('', (#4120), #29);
#4124=PROPERTY_DEFINITION_REPRESENTATION(#4122, #4123);
#4125=ORIENTED_EDGE('', *, *, #1061, .T.);
#4126=DIRECTION('', (4.551843E-16, -1.0, 0.0));
#4127=VECTOR('', #4126, 15.609999999999999);
#4128=LINE('', #1047, #4127);
#4129=EDGE_CURVE('', #1048, #4090, #4128, .T.);
#4130=ORIENTED_EDGE('', *, *, #4129, .T.);
#4131=ORIENTED_EDGE('', *, *, #4111, .F.);
#4132=DIRECTION('', (4.551843E-16, -1.0, 0.0));
#4133=VECTOR('', #4132, 15.609999999999998);
#4134=LINE('', #1056, #4133);
#4135=EDGE_CURVE('', #1057, #4100, #4134, .T.);
#4136=ORIENTED_EDGE('', *, *, #4135, .F.);
#4137=EDGE_LOOP('', (#4125, #4130, #4131, #4136));
#4138=FACE_BOUND('', #4137, .F.);
#4139=CARTESIAN_POINT('', (49.999999999999993, 0.0, 90.0));
#4140=DIRECTION('', (-1.0, -4.551843E-16, 0.0));
#4141=DIRECTION('', (4.551843E-16, -1.0, 0.0));
#4142=AXIS2_PLACEMENT_3D('', #4139, #4140, #4141);
#4143=PLANE('', #4142);
#4144=ADVANCED_FACE('', (#4138), #4143, .F.);
#4145=SHAPE_ASPECT('featured shape: face 109', $, #38, .T.);
#4146=PROPERTY_DEFINITION('', $, #4145);
#4147=FACE_SHAPE_REPRESENTATION('', (#4144), #29);
#4148=PROPERTY_DEFINITION_REPRESENTATION(#4146, #4147);
#4149=ORIENTED_EDGE('', *, *, #1077, .T.);
#4150=CARTESIAN_POINT('', (60.0, -15.610000000000003, 60.0));
#4151=VERTEX_POINT('', #4150);
#4152=DIRECTION('', (4.551843E-16, -1.0, 0.0));
#4153=VECTOR('', #4152, 15.609999999999999);
#4154=LINE('', #1063, #4153);

```

```

#4155=EDGE_CURVE('',#1064,#4151,#4154,.T.);
#4156=ORIENTED_EDGE('',*,*,#4155,.T.);
#4157=CARTESIAN_POINT('',(68.934896160916736,-12.632558739516394,60.0));
#4158=VERTEX_POINT('',#4157);
#4159=DIRECTION('',(-0.948710608132913,-0.316145823973807,0.0));
#4160=VECTOR('',#4159,9.417936391056944);
#4161=LINE('',#4157,#4160);
#4162=EDGE_CURVE('',#4158,#4151,#4161,.T.);
#4163=ORIENTED_EDGE('',*,*,#4162,.F.);
#4164=CARTESIAN_POINT('',(91.065103839083264,-12.632558739516396,60.0));
#4165=VERTEX_POINT('',#4164);
#4166=CARTESIAN_POINT('',(80.0,-45.837430024168363,60.0));
#4167=DIRECTION('',(0.0,0.0,1.000000000000000));
#4168=DIRECTION('',(0.316145823973807,0.948710608132913,0.0));
#4169=AXIS2_PLACEMENT_3D('',#4166,#4167,#4168);
#4170=CIRCLE('',#4169,35.000000000005265);
#4171=EDGE_CURVE('',#4165,#4158,#4170,.T.);
#4172=ORIENTED_EDGE('',*,*,#4171,.F.);
#4173=CARTESIAN_POINT('',(99.999999999999986,-15.610000000000001,60.0));
#4174=VERTEX_POINT('',#4173);
#4175=DIRECTION('',(-0.948710608132913,0.316145823973808,0.0));
#4176=VECTOR('',#4175,9.417936391056955);
#4177=LINE('',#4173,#4176);
#4178=EDGE_CURVE('',#4174,#4165,#4177,.T.);
#4179=ORIENTED_EDGE('',*,*,#4178,.F.);
#4180=DIRECTION('',(0.0,1.0,0.0));
#4181=VECTOR('',#4180,15.610000000000003);
#4182=LINE('',#4173,#4181);
#4183=EDGE_CURVE('',#4174,#1073,#4182,.T.);
#4184=ORIENTED_EDGE('',*,*,#4183,.T.);
#4185=EDGE_LOOP('',(#4149,#4156,#4163,#4172,#4179,#4184));
#4186=FACE_BOUND('',#4185,.F.);
#4187=CARTESIAN_POINT('',(50.0,0.0,60.0));
#4188=DIRECTION('',(0.0,0.0,1.0));
#4189=DIRECTION('',(1.0,0.0,0.0));
#4190=AXIS2_PLACEMENT_3D('',#4187,#4188,#4189);
#4191=PLANE('',#4190);
#4192=ADVANCED_FACE('',(#4186),#4191,.F.);
#4193=SHAPE_ASPECT('featured shape: face 110',$,#38,.T.);
#4194=PROPERTY_DEFINITION('',$,#4193);
#4195=FACE_SHAPE_REPRESENTATION('',(#4192),#29);
#4196=PROPERTY_DEFINITION_REPRESENTATION(#4194,#4195);
#4197=ORIENTED_EDGE('',*,*,#1054,.T.);
#4198=ORIENTED_EDGE('',*,*,#3907,.T.);
#4199=CARTESIAN_POINT('',(60.0,-15.609999999999992,20.0));
#4200=DIRECTION('',(-5.532395E-16,1.0,0.0));
#4201=DIRECTION('',(0.0,0.0,-1.0));
#4202=AXIS2_PLACEMENT_3D('',#4199,#4200,#4201);
#4203=CIRCLE('',#4202,10.0);
#4204=EDGE_CURVE('',#3898,#4090,#4203,.T.);
#4205=ORIENTED_EDGE('',*,*,#4204,.T.);
#4206=ORIENTED_EDGE('',*,*,#4129,.F.);
#4207=EDGE_LOOP('',(#4197,#4198,#4205,#4206));

```

```

#4208=FACE_BOUND('',#4207,.F.);
#4209=CARTESIAN_POINT('',(60.000000000000007,-25.643252392135473,20.0));
#4210=DIRECTION('',(-5.532395E-16,1.0,0.0));
#4211=DIRECTION('',(0.0,0.0,-1.0));
#4212=AXIS2_PLACEMENT_3D('',#4209,#4210,#4211);
#4213=CYLINDRICAL_SURFACE('',#4212,10.0);
#4214=ADVANCED_FACE('',(#4208),#4213,.F.);
#4215=SHAPE_ASPECT('featured shape: face 111',$,#38,.T.);
#4216=PROPERTY_DEFINITION('',$,#4215);
#4217=FACE_SHAPE_REPRESENTATION('',(#4214),#29);
#4218=PROPERTY_DEFINITION_REPRESENTATION(#4216,#4217);
#4219=ORIENTED_EDGE('',*,*,#3902,.F.);
#4220=CARTESIAN_POINT('',(68.934896160916736,-12.632558739516401,20.0));
#4221=DIRECTION('',(-0.948710608132914,-0.316145823973807,-1.684763E-16));
#4222=DIRECTION('',(0.0,5.329071E-16,-1.0));
#4223=AXIS2_PLACEMENT_3D('',#4220,#4221,#4222);
#4224=CIRCLE('',#4223,10.0);
#4225=EDGE_CURVE('',#3889,#4022,#4224,.T.);
#4226=ORIENTED_EDGE('',*,*,#4225,.T.);
#4227=ORIENTED_EDGE('',*,*,#4062,.T.);
#4228=CARTESIAN_POINT('',(60.000000000000043,-15.609999999999989,20.0));
#4229=DIRECTION('',(-0.948710608132914,-0.316145823973807,1.123176E-16));
#4230=DIRECTION('',(0.0,-3.552714E-16,-1.0));
#4231=AXIS2_PLACEMENT_3D('',#4228,#4229,#4230);
#4232=CIRCLE('',#4231,10.0);
#4233=EDGE_CURVE('',#3898,#4058,#4232,.T.);
#4234=ORIENTED_EDGE('',*,*,#4233,.F.);
#4235=EDGE_LOOP('',(#4219,#4226,#4227,#4234));
#4236=FACE_BOUND('',#4235,.F.);
#4237=CARTESIAN_POINT('',(106.973637467630653,0.043371212405010,20.0));
#4238=DIRECTION('',(-0.948710608132914,-0.316145823973807,0.0));
#4239=DIRECTION('',(0.0,0.0,-1.0));
#4240=AXIS2_PLACEMENT_3D('',#4237,#4238,#4239);
#4241=CYLINDRICAL_SURFACE('',#4240,10.0);
#4242=ADVANCED_FACE('',(#4236),#4241,.F.);
#4243=SHAPE_ASPECT('featured shape: face 112',$,#38,.T.);
#4244=PROPERTY_DEFINITION('',$,#4243);
#4245=FACE_SHAPE_REPRESENTATION('',(#4242),#29);
#4246=PROPERTY_DEFINITION_REPRESENTATION(#4244,#4245);
#4247=ORIENTED_EDGE('',*,*,#3895,.F.);
#4248=CARTESIAN_POINT('',(91.065103839083250,-12.632558739516391,20.0));
#4249=DIRECTION('',(-0.948710608132914,0.316145823973807,0.0));
#4250=DIRECTION('',(0.0,0.0,-1.0));
#4251=AXIS2_PLACEMENT_3D('',#4248,#4249,#4250);
#4252=CIRCLE('',#4251,10.0);
#4253=EDGE_CURVE('',#3882,#3990,#4252,.T.);
#4254=ORIENTED_EDGE('',*,*,#4253,.T.);
#4255=ORIENTED_EDGE('',*,*,#4028,.T.);
#4256=ORIENTED_EDGE('',*,*,#4225,.F.);
#4257=EDGE_LOOP('',(#4247,#4254,#4255,#4256));
#4258=FACE_BOUND('',#4257,.F.);
#4259=CARTESIAN_POINT('',(80.0,-45.837430024168363,20.0));
#4260=DIRECTION('',(0.0,0.0,1.0));

```



```

#4261=DIRECTION('', (0.316336760367410,0.948646959643181,0.0));
#4262=AXIS2_PLACEMENT_3D('', #4259, #4260, #4261);
#4263=TOROIDAL_SURFACE('', #4262, 35.0, 10.0);
#4264=ADVANCED_FACE('', (#4258), #4263, .F.);
#4265=SHAPE_ASPECT('featured shape: face 113', $, #38, .T.);
#4266=PROPERTY_DEFINITION('', $, #4265);
#4267=FACE_SHAPE_REPRESENTATION('', (#4264), #29);
#4268=PROPERTY_DEFINITION_REPRESENTATION(#4266, #4267);
#4269=ORIENTED_EDGE('', *, *, #3994, .T.);
#4270=ORIENTED_EDGE('', *, *, #4253, .F.);
#4271=ORIENTED_EDGE('', *, *, #3886, .F.);
#4272=CARTESIAN_POINT('', (99.999999999999986, -15.610000000000003, 20.0));
#4273=DIRECTION('', (-0.948710608132914, 0.316145823973807, 1.684763E-16));
#4274=DIRECTION('', (0.0, 5.329071E-16, -1.0));
#4275=AXIS2_PLACEMENT_3D('', #4272, #4273, #4274);
#4276=CIRCLE('', #4275, 10.0);
#4277=EDGE_CURVE('', #3875, #3954, #4276, .T.);
#4278=ORIENTED_EDGE('', *, *, #4277, .T.);
#4279=EDGE_LOOP('', (#4269, #4270, #4271, #4278));
#4280=FACE_BOUND('', #4279, .F.);
#4281=CARTESIAN_POINT('', (110.060467428838379, -18.962523664841502, 20.0));
#4282=DIRECTION('', (-0.948710608132914, 0.316145823973807, 0.0));
#4283=DIRECTION('', (0.0, 0.0, -1.0));
#4284=AXIS2_PLACEMENT_3D('', #4281, #4282, #4283);
#4285=CYLINDRICAL_SURFACE('', #4284, 10.0);
#4286=ADVANCED_FACE('', (#4280), #4285, .F.);
#4287=SHAPE_ASPECT('featured shape: face 114', $, #38, .T.);
#4288=PROPERTY_DEFINITION('', $, #4287);
#4289=FACE_SHAPE_REPRESENTATION('', (#4286), #29);
#4290=PROPERTY_DEFINITION_REPRESENTATION(#4288, #4289);
#4291=ORIENTED_EDGE('', *, *, #1100, .F.);
#4292=ORIENTED_EDGE('', *, *, #3939, .F.);
#4293=CARTESIAN_POINT('', (99.999999999999986, -15.609999999999994, 20.0));
#4294=DIRECTION('', (0.0, -1.0, 0.0));
#4295=DIRECTION('', (0.0, 0.0, -1.0));
#4296=AXIS2_PLACEMENT_3D('', #4293, #4294, #4295);
#4297=CIRCLE('', #4296, 10.0);
#4298=EDGE_CURVE('', #3875, #3930, #4297, .T.);
#4299=ORIENTED_EDGE('', *, *, #4298, .F.);
#4300=ORIENTED_EDGE('', *, *, #3879, .T.);
#4301=EDGE_LOOP('', (#4291, #4292, #4299, #4300));
#4302=FACE_BOUND('', #4301, .F.);
#4303=CARTESIAN_POINT('', (99.999999999999986, 0.043371212405010, 20.0));
#4304=DIRECTION('', (0.0, -1.0, 0.0));
#4305=DIRECTION('', (0.0, 0.0, -1.0));
#4306=AXIS2_PLACEMENT_3D('', #4303, #4304, #4305);
#4307=CYLINDRICAL_SURFACE('', #4306, 10.0);
#4308=ADVANCED_FACE('', (#4302), #4307, .F.);
#4309=SHAPE_ASPECT('featured shape: face 115', $, #38, .T.);
#4310=PROPERTY_DEFINITION('', $, #4309);
#4311=FACE_SHAPE_REPRESENTATION('', (#4308), #29);
#4312=PROPERTY_DEFINITION_REPRESENTATION(#4310, #4311);
#4313=ORIENTED_EDGE('', *, *, #3960, .T.);

```

```

#4314=ORIENTED_EDGE('',*,*,#4277,.F.);
#4315=ORIENTED_EDGE('',*,*,#4298,.T.);
#4316=EDGE_LOOP('',(#4313,#4314,#4315));
#4317=FACE_BOUND('',#4316,.F.);
#4318=CARTESIAN_POINT('',(99.999999999999986,-15.6100000000000003,20.0));
#4319=DIRECTION('',(0.223548855975687,0.670839704394397,-0.707106781186547));
#4320=DIRECTION('',(-0.223548855975687,-0.670839704394397,-
0.707106781186548));
#4321=AXIS2_PLACEMENT_3D('',#4318,#4319,#4320);
#4322=SPHERICAL_SURFACE('',#4321,10.0);
#4323=ADVANCED_FACE('',(#4317),#4322,.F.);
#4324=SHAPE_ASPECT('featured shape: face 116',$,#38,.T.);
#4325=PROPERTY_DEFINITION('',$,#4324);
#4326=FACE_SHAPE_REPRESENTATION('',(#4323),#29);
#4327=PROPERTY_DEFINITION_REPRESENTATION(#4325,#4326);
#4328=ORIENTED_EDGE('',*,*,#4096,.T.);
#4329=ORIENTED_EDGE('',*,*,#4204,.F.);
#4330=ORIENTED_EDGE('',*,*,#4233,.T.);
#4331=EDGE_LOOP('',(#4328,#4329,#4330));
#4332=FACE_BOUND('',#4331,.F.);
#4333=CARTESIAN_POINT('',(60.0,-15.6100000000000001,20.0));
#4334=DIRECTION('',(0.707106781186547,3.768222E-16,-0.707106781186548));
#4335=DIRECTION('',(-0.707106781186547,-3.768222E-16,-0.707106781186547));
#4336=AXIS2_PLACEMENT_3D('',#4333,#4334,#4335);
#4337=SPHERICAL_SURFACE('',#4336,10.0);
#4338=ADVANCED_FACE('',(#4332),#4337,.F.);
#4339=SHAPE_ASPECT('featured shape: face 117',$,#38,.T.);
#4340=PROPERTY_DEFINITION('',$,#4339);
#4341=FACE_SHAPE_REPRESENTATION('',(#4338),#29);
#4342=PROPERTY_DEFINITION_REPRESENTATION(#4340,#4341);
#4343=ORIENTED_EDGE('',*,*,#1070,.T.);
#4344=ORIENTED_EDGE('',*,*,#4135,.T.);
#4345=CARTESIAN_POINT('',(60.0,-15.6100000000000007,50.0));
#4346=DIRECTION('',(-5.329071E-16,1.0,0.0));
#4347=DIRECTION('',(-1.0,-5.329071E-16,0.0));
#4348=AXIS2_PLACEMENT_3D('',#4345,#4346,#4347);
#4349=CIRCLE('',#4348,10.0);
#4350=EDGE_CURVE('',#4100,#4151,#4349,.T.);
#4351=ORIENTED_EDGE('',*,*,#4350,.T.);
#4352=ORIENTED_EDGE('',*,*,#4155,.F.);
#4353=EDGE_LOOP('',(#4343,#4344,#4351,#4352));
#4354=FACE_BOUND('',#4353,.F.);
#4355=CARTESIAN_POINT('',(60.000000000000007,-25.649428979163300,50.0));
#4356=DIRECTION('',(-5.533360E-16,1.0,0.0));
#4357=DIRECTION('',(-1.0,-5.533360E-16,0.0));
#4358=AXIS2_PLACEMENT_3D('',#4355,#4356,#4357);
#4359=CYLINDRICAL_SURFACE('',#4358,10.0);
#4360=ADVANCED_FACE('',(#4354),#4359,.F.);
#4361=SHAPE_ASPECT('featured shape: face 118',$,#38,.T.);
#4362=PROPERTY_DEFINITION('',$,#4361);
#4363=FACE_SHAPE_REPRESENTATION('',(#4360),#29);
#4364=PROPERTY_DEFINITION_REPRESENTATION(#4362,#4363);
#4365=ORIENTED_EDGE('',*,*,#4070,.F.);

```

```

#4366=CARTESIAN_POINT('', (68.934896160916765, -12.632558739516384, 50.0));
#4367=DIRECTION('', (-0.948710608132914, -0.316145823973807, 0.0));
#4368=DIRECTION('', (0.316145823973807, -0.948710608132914, 0.0));
#4369=AXIS2_PLACEMENT_3D('', #4366, #4367, #4368);
#4370=CIRCLE('', #4369, 10.0);
#4371=EDGE_CURVE('', #4032, #4158, #4370, .T.);
#4372=ORIENTED_EDGE('', *, *, #4371, .T.);
#4373=ORIENTED_EDGE('', *, *, #4162, .T.);
#4374=CARTESIAN_POINT('', (59.999999999999957, -15.610000000000014, 50.0));
#4375=DIRECTION('', (-0.948710608132913, -0.316145823973807, 0.0));
#4376=DIRECTION('', (0.316145823973807, -0.948710608132913, 0.0));
#4377=AXIS2_PLACEMENT_3D('', #4374, #4375, #4376);
#4378=CIRCLE('', #4377, 10.0);
#4379=EDGE_CURVE('', #4066, #4151, #4378, .T.);
#4380=ORIENTED_EDGE('', *, *, #4379, .F.);
#4381=EDGE_LOOP('', (#4365, #4372, #4373, #4380));
#4382=FACE_BOUND('', #4381, .F.);
#4383=CARTESIAN_POINT('', (106.941654523051668, 0.032713300202545, 50.0));
#4384=DIRECTION('', (-0.948710608132914, -0.316145823973807, 0.0));
#4385=DIRECTION('', (0.316145823973807, -0.948710608132914, 0.0));
#4386=AXIS2_PLACEMENT_3D('', #4383, #4384, #4385);
#4387=CYLINDRICAL_SURFACE('', #4386, 10.0);
#4388=ADVANCED_FACE('', (#4382), #4387, .F.);
#4389=SHAPE_ASPECT('featured shape: face 119', $, #38, .T.);
#4390=PROPERTY_DEFINITION('', $, #4389);
#4391=FACE_SHAPE_REPRESENTATION('', (#4388), #29);
#4392=PROPERTY_DEFINITION_REPRESENTATION(#4390, #4391);
#4393=ORIENTED_EDGE('', *, *, #4038, .F.);
#4394=CARTESIAN_POINT('', (91.065103839083264, -12.632558739516391, 50.0));
#4395=DIRECTION('', (-0.948710608132914, 0.316145823973807, 0.0));
#4396=DIRECTION('', (-0.316145823973807, -0.948710608132914, 0.0));
#4397=AXIS2_PLACEMENT_3D('', #4394, #4395, #4396);
#4398=CIRCLE('', #4397, 10.0);
#4399=EDGE_CURVE('', #3998, #4165, #4398, .T.);
#4400=ORIENTED_EDGE('', *, *, #4399, .T.);
#4401=ORIENTED_EDGE('', *, *, #4171, .T.);
#4402=ORIENTED_EDGE('', *, *, #4371, .F.);
#4403=EDGE_LOOP('', (#4393, #4400, #4401, #4402));
#4404=FACE_BOUND('', #4403, .F.);
#4405=CARTESIAN_POINT('', (80.0, -45.837430024168363, 50.0));
#4406=DIRECTION('', (0.0, 0.0, 1.0));
#4407=DIRECTION('', (0.316345968619440, 0.948643889000624, 0.0));
#4408=AXIS2_PLACEMENT_3D('', #4405, #4406, #4407);
#4409=TOROIDAL_SURFACE('', #4408, 34.999999999999993, 10.0);
#4410=ADVANCED_FACE('', (#4404), #4409, .F.);
#4411=SHAPE_ASPECT('featured shape: face 120', $, #38, .T.);
#4412=PROPERTY_DEFINITION('', $, #4411);
#4413=FACE_SHAPE_REPRESENTATION('', (#4410), #29);
#4414=PROPERTY_DEFINITION_REPRESENTATION(#4412, #4413);
#4415=ORIENTED_EDGE('', *, *, #4178, .T.);
#4416=ORIENTED_EDGE('', *, *, #4399, .F.);
#4417=ORIENTED_EDGE('', *, *, #4002, .F.);
#4418=CARTESIAN_POINT('', (100.0, -15.610000000000007, 50.0));

```

```

#4419=DIRECTION('', (-0.948710608132914, 0.316145823973807, 0.0));
#4420=DIRECTION('', (-0.316145823973807, -0.948710608132914, 0.0));
#4421=AXIS2_PLACEMENT_3D('', #4418, #4419, #4420);
#4422=CIRCLE('', #4421, 10.0);
#4423=EDGE_CURVE('', #3964, #4174, #4422, .T.);
#4424=ORIENTED_EDGE('', *, *, #4423, .T.);
#4425=EDGE_LOOP('', (#4415, #4416, #4417, #4424));
#4426=FACE_BOUND('', #4425, .F.);
#4427=CARTESIAN_POINT('', (110.080242558619432, -18.969113476998370, 50.0));
#4428=DIRECTION('', (-0.948710608132914, 0.316145823973807, 0.0));
#4429=DIRECTION('', (-0.316145823973807, -0.948710608132914, 0.0));
#4430=AXIS2_PLACEMENT_3D('', #4427, #4428, #4429);
#4431=CYLINDRICAL_SURFACE('', #4430, 10.0);
#4432=ADVANCED_FACE('', (#4426), #4431, .F.);
#4433=SHAPE_ASPECT('featured shape: face 121', $, #38, .T.);
#4434=PROPERTY_DEFINITION('', $, #4433);
#4435=FACE_SHAPE_REPRESENTATION('', (#4432), #29);
#4436=PROPERTY_DEFINITION_REPRESENTATION(#4434, #4435);
#4437=ORIENTED_EDGE('', *, *, #1086, .F.);
#4438=ORIENTED_EDGE('', *, *, #4183, .F.);
#4439=CARTESIAN_POINT('', (99.999999999999986, -15.610000000000010, 50.0));
#4440=DIRECTION('', (0.0, -1.0, 0.0));
#4441=DIRECTION('', (1.0, 0.0, 0.0));
#4442=AXIS2_PLACEMENT_3D('', #4439, #4440, #4441);
#4443=CIRCLE('', #4442, 10.0);
#4444=EDGE_CURVE('', #3923, #4174, #4443, .T.);
#4445=ORIENTED_EDGE('', *, *, #4444, .F.);
#4446=ORIENTED_EDGE('', *, *, #3927, .T.);
#4447=EDGE_LOOP('', (#4437, #4438, #4445, #4446));
#4448=FACE_BOUND('', #4447, .F.);
#4449=CARTESIAN_POINT('', (99.999999999999986, 0.032713300202545, 50.0));
#4450=DIRECTION('', (0.0, -1.0, 0.0));
#4451=DIRECTION('', (1.0, 0.0, 0.0));
#4452=AXIS2_PLACEMENT_3D('', #4449, #4450, #4451);
#4453=CYLINDRICAL_SURFACE('', #4452, 10.0);
#4454=ADVANCED_FACE('', (#4448), #4453, .F.);
#4455=SHAPE_ASPECT('featured shape: face 122', $, #38, .T.);
#4456=PROPERTY_DEFINITION('', $, #4455);
#4457=FACE_SHAPE_REPRESENTATION('', (#4454), #29);
#4458=PROPERTY_DEFINITION_REPRESENTATION(#4456, #4457);
#4459=ORIENTED_EDGE('', *, *, #3970, .T.);
#4460=ORIENTED_EDGE('', *, *, #4444, .T.);
#4461=ORIENTED_EDGE('', *, *, #4423, .F.);
#4462=EDGE_LOOP('', (#4459, #4460, #4461));
#4463=FACE_BOUND('', #4462, .F.);
#4464=CARTESIAN_POINT('', (99.999999999999986, -15.610000000000003, 50.0));
#4465=DIRECTION('', (-0.223548855975687, -0.670839704394397, -
0.707106781186547));
#4466=DIRECTION('', (-0.223548855975687, -
0.670839704394397, 0.707106781186548));
#4467=AXIS2_PLACEMENT_3D('', #4464, #4465, #4466);
#4468=SPHERICAL_SURFACE('', #4467, 10.0);
#4469=ADVANCED_FACE('', (#4463), #4468, .F.);

```

```

#4470=SHAPE_ASPECT('featured shape: face 123', $, #38, .T.);
#4471=PROPERTY_DEFINITION('', $, #4470);
#4472=FACE_SHAPE_REPRESENTATION('', (#4469), #29);
#4473=PROPERTY_DEFINITION_REPRESENTATION(#4471, #4472);
#4474=ORIENTED_EDGE('', *, *, #4106, .T.);
#4475=ORIENTED_EDGE('', *, *, #4379, .T.);
#4476=ORIENTED_EDGE('', *, *, #4350, .F.);
#4477=EDGE_LOOP('', (#4474, #4475, #4476));
#4478=FACE_BOUND('', #4477, .F.);
#4479=CARTESIAN_POINT('', (60.0, -15.609999999999999, 50.0));
#4480=DIRECTION('', (-0.707106781186548, -3.768222E-16, -0.707106781186547));
#4481=DIRECTION('', (-0.707106781186547, -3.768222E-16, 0.707106781186547));
#4482=AXIS2_PLACEMENT_3D('', #4479, #4480, #4481);
#4483=SPHERICAL_SURFACE('', #4482, 10.0);
#4484=ADVANCED_FACE('', (#4478), #4483, .F.);
#4485=SHAPE_ASPECT('featured shape: face 124', $, #38, .T.);
#4486=PROPERTY_DEFINITION('', $, #4485);
#4487=FACE_SHAPE_REPRESENTATION('', (#4484), #29);
#4488=PROPERTY_DEFINITION_REPRESENTATION(#4486, #4487);
#4489=CARTESIAN_POINT('', (50.0, -10.0, 90.0));
#4490=DIRECTION('', (0.0, -1.0, 0.0));
#4491=DIRECTION('', (0.0, 0.0, -1.0));
#4492=AXIS2_PLACEMENT_3D('', #4489, #4490, #4491);
#4493=CIRCLE('', #4492, 9.0);
#4494=EDGE_CURVE('', #2090, #2064, #4493, .T.);
#4495=ORIENTED_EDGE('', *, *, #4494, .F.);
#4496=ORIENTED_EDGE('', *, *, #2101, .F.);
#4497=ORIENTED_EDGE('', *, *, #712, .T.);
#4498=ORIENTED_EDGE('', *, *, #2073, .T.);
#4499=EDGE_LOOP('', (#4495, #4496, #4497, #4498));
#4500=FACE_BOUND('', #4499, .F.);
#4501=CARTESIAN_POINT('', (50.0, 0.0, 90.0));
#4502=DIRECTION('', (0.0, -1.0, 0.0));
#4503=DIRECTION('', (1.0, 0.0, 0.0));
#4504=AXIS2_PLACEMENT_3D('', #4501, #4502, #4503);
#4505=CYLINDRICAL_SURFACE('', #4504, 9.0);
#4506=ADVANCED_FACE('', (#4500), #4505, .F.);
#4507=SHAPE_ASPECT('featured shape: face 125', $, #38, .T.);
#4508=PROPERTY_DEFINITION('', $, #4507);
#4509=FACE_SHAPE_REPRESENTATION('', (#4506), #29);
#4510=PROPERTY_DEFINITION_REPRESENTATION(#4508, #4509);
#4511=CARTESIAN_POINT('', (47.9999999999999979, -20.0, 135.0));
#4512=VERTEX_POINT('', #4511);
#4513=DIRECTION('', (0.0, 0.0, 1.0));
#4514=VECTOR('', #4513, 45.0);
#4515=LINE('', #4511, #4514);
#4516=EDGE_CURVE('', #4512, #80, #4515, .T.);
#4517=ORIENTED_EDGE('', *, *, #4516, .F.);
#4518=CARTESIAN_POINT('', (52.9999999999999979, -20.0, 130.0));
#4519=VERTEX_POINT('', #4518);
#4520=CARTESIAN_POINT('', (52.9999999999999979, -20.0, 135.0));
#4521=DIRECTION('', (0.0, 1.0, 0.0));
#4522=DIRECTION('', (0.0, 0.0, -1.0));

```

```

#4523=AXIS2_PLACEMENT_3D('', #4520, #4521, #4522);
#4524=CIRCLE('', #4523, 5.0);
#4525=EDGE_CURVE('', #4519, #4512, #4524, .T.);
#4526=ORIENTED_EDGE('', *, *, #4525, .F.);
#4527=CARTESIAN_POINT('', (112.999999999999943, -20.0, 130.0));
#4528=VERTEX_POINT('', #4527);
#4529=DIRECTION('', (-1.0, 0.0, 0.0));
#4530=VECTOR('', #4529, 59.999999999999964);
#4531=LINE('', #4527, #4530);
#4532=EDGE_CURVE('', #4528, #4519, #4531, .T.);
#4533=ORIENTED_EDGE('', *, *, #4532, .F.);
#4534=CARTESIAN_POINT('', (117.999999999999943, -20.0, 135.0));
#4535=VERTEX_POINT('', #4534);
#4536=CARTESIAN_POINT('', (112.999999999999943, -20.0, 135.0));
#4537=DIRECTION('', (0.0, 1.0, 0.0));
#4538=DIRECTION('', (1.0, 0.0, 0.0));
#4539=AXIS2_PLACEMENT_3D('', #4536, #4537, #4538);
#4540=CIRCLE('', #4539, 5.0);
#4541=EDGE_CURVE('', #4535, #4528, #4540, .T.);
#4542=ORIENTED_EDGE('', *, *, #4541, .F.);
#4543=DIRECTION('', (0.0, 0.0, -1.0));
#4544=VECTOR('', #4543, 45.0);
#4545=LINE('', #81, #4544);
#4546=EDGE_CURVE('', #82, #4535, #4545, .T.);
#4547=ORIENTED_EDGE('', *, *, #4546, .F.);
#4548=ORIENTED_EDGE('', *, *, #86, .F.);
#4549=EDGE_LOOP('', (#4517, #4526, #4533, #4542, #4547, #4548));
#4550=FACE_BOUND('', #4549, .F.);
#4551=CARTESIAN_POINT('', (69.999999999999972, -20.0, 145.0));
#4552=VERTEX_POINT('', #4551);
#4553=CARTESIAN_POINT('', (95.999999999999972, -20.0, 145.0));
#4554=VERTEX_POINT('', #4553);
#4555=DIRECTION('', (1.0, 0.0, 0.0));
#4556=VECTOR('', #4555, 26.0);
#4557=LINE('', #4551, #4556);
#4558=EDGE_CURVE('', #4552, #4554, #4557, .T.);
#4559=ORIENTED_EDGE('', *, *, #4558, .F.);
#4560=CARTESIAN_POINT('', (67.999999999999972, -20.0, 147.0));
#4561=VERTEX_POINT('', #4560);
#4562=CARTESIAN_POINT('', (69.999999999999972, -20.0, 147.0));
#4563=DIRECTION('', (0.0, -1.0, 0.0));
#4564=DIRECTION('', (-1.0, 0.0, 0.0));
#4565=AXIS2_PLACEMENT_3D('', #4562, #4563, #4564);
#4566=CIRCLE('', #4565, 2.0);
#4567=EDGE_CURVE('', #4561, #4552, #4566, .T.);
#4568=ORIENTED_EDGE('', *, *, #4567, .F.);
#4569=CARTESIAN_POINT('', (67.999999999999972, -20.0, 163.0));
#4570=VERTEX_POINT('', #4569);
#4571=DIRECTION('', (0.0, 0.0, -1.0));
#4572=VECTOR('', #4571, 16.0);
#4573=LINE('', #4569, #4572);
#4574=EDGE_CURVE('', #4570, #4561, #4573, .T.);
#4575=ORIENTED_EDGE('', *, *, #4574, .F.);

```

```

#4576=CARTESIAN_POINT('', (69.999999999999972, -20.0, 165.0));
#4577=VERTEX_POINT('', #4576);
#4578=CARTESIAN_POINT('', (69.999999999999972, -20.0, 163.0));
#4579=DIRECTION('', (0.0, -1.0, 0.0));
#4580=DIRECTION('', (0.0, 0.0, 1.0));
#4581=AXIS2_PLACEMENT_3D('', #4578, #4579, #4580);
#4582=CIRCLE('', #4581, 2.0);
#4583=EDGE_CURVE('', #4577, #4570, #4582, .T.);
#4584=ORIENTED_EDGE('', *, *, #4583, .F.);
#4585=CARTESIAN_POINT('', (95.999999999999972, -20.0, 165.0));
#4586=VERTEX_POINT('', #4585);
#4587=DIRECTION('', (-1.0, 0.0, 0.0));
#4588=VECTOR('', #4587, 26.0);
#4589=LINE('', #4585, #4588);
#4590=EDGE_CURVE('', #4586, #4577, #4589, .T.);
#4591=ORIENTED_EDGE('', *, *, #4590, .F.);
#4592=CARTESIAN_POINT('', (97.999999999999972, -20.0, 163.0));
#4593=VERTEX_POINT('', #4592);
#4594=CARTESIAN_POINT('', (95.999999999999972, -20.0, 163.0));
#4595=DIRECTION('', (0.0, -1.0, 0.0));
#4596=DIRECTION('', (1.0, 0.0, 0.0));
#4597=AXIS2_PLACEMENT_3D('', #4594, #4595, #4596);
#4598=CIRCLE('', #4597, 2.0);
#4599=EDGE_CURVE('', #4593, #4586, #4598, .T.);
#4600=ORIENTED_EDGE('', *, *, #4599, .F.);
#4601=CARTESIAN_POINT('', (97.999999999999972, -20.0, 147.0));
#4602=VERTEX_POINT('', #4601);
#4603=DIRECTION('', (0.0, 0.0, 1.0));
#4604=VECTOR('', #4603, 16.0);
#4605=LINE('', #4601, #4604);
#4606=EDGE_CURVE('', #4602, #4593, #4605, .T.);
#4607=ORIENTED_EDGE('', *, *, #4606, .F.);
#4608=CARTESIAN_POINT('', (95.999999999999972, -20.0, 147.0));
#4609=DIRECTION('', (0.0, -1.0, 0.0));
#4610=DIRECTION('', (0.0, 0.0, -1.0));
#4611=AXIS2_PLACEMENT_3D('', #4608, #4609, #4610);
#4612=CIRCLE('', #4611, 2.0);
#4613=EDGE_CURVE('', #4554, #4602, #4612, .T.);
#4614=ORIENTED_EDGE('', *, *, #4613, .F.);
#4615=EDGE_LOOP('', (#4559, #4568, #4575, #4584, #4591, #4600, #4607, #4614));
#4616=FACE_BOUND('', #4615, .F.);
#4617=CARTESIAN_POINT('', (0.0, -20.0, 180.0));
#4618=DIRECTION('', (0.0, 1.0, 0.0));
#4619=DIRECTION('', (1.0, 0.0, 0.0));
#4620=AXIS2_PLACEMENT_3D('', #4617, #4618, #4619);
#4621=PLANE('', #4620);
#4622=ADVANCED_FACE('', (#4550, #4616), #4621, .T.);
#4623=SHAPE_ASPECT('featured shape: face 126', $, #38, .T.);
#4624=PROPERTY_DEFINITION('', $, #4623);
#4625=FACE_SHAPE_REPRESENTATION('', (#4622), #29);
#4626=PROPERTY_DEFINITION_REPRESENTATION(#4624, #4625);
#4627=ORIENTED_EDGE('', *, *, #93, .T.);
#4628=ORIENTED_EDGE('', *, *, #4546, .T.);

```

```

#4629=DIRECTION('', (0.0, -1.0, 0.0));
#4630=VECTOR('', #4629, 20.0);
#4631=LINE('', #652, #4630);
#4632=EDGE_CURVE('', #653, #4535, #4631, .T.);
#4633=ORIENTED_EDGE('', *, *, #4632, .F.);
#4634=ORIENTED_EDGE('', *, *, #657, .F.);
#4635=EDGE_LOOP('', (#4627, #4628, #4633, #4634));
#4636=FACE_BOUND('', #4635, .F.);
#4637=CARTESIAN_POINT('', (117.999999999999943, 0.0, 180.0));
#4638=DIRECTION('', (1.0, 0.0, 0.0));
#4639=DIRECTION('', (0.0, 0.0, -1.0));
#4640=AXIS2_PLACEMENT_3D('', #4637, #4638, #4639);
#4641=PLANE('', #4640);
#4642=ADVANCED_FACE('', (#4636), #4641, .F.);
#4643=SHAPE_ASPECT('featured shape: face 127', $, #38, .T.);
#4644=PROPERTY_DEFINITION('', $, #4643);
#4645=FACE_SHAPE_REPRESENTATION('', (#4642), #29);
#4646=PROPERTY_DEFINITION_REPRESENTATION(#4644, #4645);
#4647=ORIENTED_EDGE('', *, *, #666, .F.);
#4648=ORIENTED_EDGE('', *, *, #4632, .T.);
#4649=ORIENTED_EDGE('', *, *, #4541, .T.);
#4650=DIRECTION('', (0.0, -1.0, 0.0));
#4651=VECTOR('', #4650, 20.0);
#4652=LINE('', #659, #4651);
#4653=EDGE_CURVE('', #660, #4528, #4652, .T.);
#4654=ORIENTED_EDGE('', *, *, #4653, .F.);
#4655=EDGE_LOOP('', (#4647, #4648, #4649, #4654));
#4656=FACE_BOUND('', #4655, .F.);
#4657=CARTESIAN_POINT('', (112.999999999999943, 0.0, 135.0));
#4658=DIRECTION('', (0.0, 1.0, 0.0));
#4659=DIRECTION('', (1.0, 0.0, 0.0));
#4660=AXIS2_PLACEMENT_3D('', #4657, #4658, #4659);
#4661=CYLINDRICAL_SURFACE('', #4660, 5.0);
#4662=ADVANCED_FACE('', (#4656), #4661, .F.);
#4663=SHAPE_ASPECT('featured shape: face 128', $, #38, .T.);
#4664=PROPERTY_DEFINITION('', $, #4663);
#4665=FACE_SHAPE_REPRESENTATION('', (#4662), #29);
#4666=PROPERTY_DEFINITION_REPRESENTATION(#4664, #4665);
#4667=ORIENTED_EDGE('', *, *, #673, .F.);
#4668=ORIENTED_EDGE('', *, *, #4653, .T.);
#4669=ORIENTED_EDGE('', *, *, #4532, .T.);
#4670=DIRECTION('', (0.0, -1.0, 0.0));
#4671=VECTOR('', #4670, 20.0);
#4672=LINE('', #668, #4671);
#4673=EDGE_CURVE('', #669, #4519, #4672, .T.);
#4674=ORIENTED_EDGE('', *, *, #4673, .F.);
#4675=EDGE_LOOP('', (#4667, #4668, #4669, #4674));
#4676=FACE_BOUND('', #4675, .F.);
#4677=CARTESIAN_POINT('', (112.999999999999943, 0.0, 130.0));
#4678=DIRECTION('', (0.0, 0.0, -1.0));
#4679=DIRECTION('', (-1.0, 0.0, 0.0));
#4680=AXIS2_PLACEMENT_3D('', #4677, #4678, #4679);
#4681=PLANE('', #4680);

```



```

#4682=ADVANCED_FACE('', (#4676), #4681, .F.);
#4683=SHAPE_ASPECT('featured shape: face 129', $, #38, .T.);
#4684=PROPERTY_DEFINITION('', $, #4683);
#4685=FACE_SHAPE_REPRESENTATION('', (#4682), #29);
#4686=PROPERTY_DEFINITION_REPRESENTATION(#4684, #4685);
#4687=ORIENTED_EDGE('', *, *, #682, .F.);
#4688=ORIENTED_EDGE('', *, *, #4673, .T.);
#4689=ORIENTED_EDGE('', *, *, #4525, .T.);
#4690=DIRECTION('', (0.0, -1.0, 0.0));
#4691=VECTOR('', #4690, 20.0);
#4692=LINE('', #675, #4691);
#4693=EDGE_CURVE('', #676, #4512, #4692, .T.);
#4694=ORIENTED_EDGE('', *, *, #4693, .F.);
#4695=EDGE_LOOP('', (#4687, #4688, #4689, #4694));
#4696=FACE_BOUND('', #4695, .F.);
#4697=CARTESIAN_POINT('', (52.9999999999999979, 0.0, 135.0));
#4698=DIRECTION('', (0.0, 1.0, 0.0));
#4699=DIRECTION('', (1.0, 0.0, 0.0));
#4700=AXIS2_PLACEMENT_3D('', #4697, #4698, #4699);
#4701=CYLINDRICAL_SURFACE('', #4700, 5.0);
#4702=ADVANCED_FACE('', (#4696), #4701, .F.);
#4703=SHAPE_ASPECT('featured shape: face 130', $, #38, .T.);
#4704=PROPERTY_DEFINITION('', $, #4703);
#4705=FACE_SHAPE_REPRESENTATION('', (#4702), #29);
#4706=PROPERTY_DEFINITION_REPRESENTATION(#4704, #4705);
#4707=ORIENTED_EDGE('', *, *, #133, .F.);
#4708=ORIENTED_EDGE('', *, *, #687, .T.);
#4709=ORIENTED_EDGE('', *, *, #4693, .T.);
#4710=ORIENTED_EDGE('', *, *, #4516, .T.);
#4711=EDGE_LOOP('', (#4707, #4708, #4709, #4710));
#4712=FACE_BOUND('', #4711, .F.);
#4713=CARTESIAN_POINT('', (47.9999999999999979, 0.0, 135.0));
#4714=DIRECTION('', (-1.0, 0.0, 0.0));
#4715=DIRECTION('', (0.0, 0.0, 1.0));
#4716=AXIS2_PLACEMENT_3D('', #4713, #4714, #4715);
#4717=PLANE('', #4716);
#4718=ADVANCED_FACE('', (#4712), #4717, .F.);
#4719=SHAPE_ASPECT('featured shape: face 131', $, #38, .T.);
#4720=PROPERTY_DEFINITION('', $, #4719);
#4721=FACE_SHAPE_REPRESENTATION('', (#4718), #29);
#4722=PROPERTY_DEFINITION_REPRESENTATION(#4720, #4721);
#4723=ORIENTED_EDGE('', *, *, #1121, .F.);
#4724=DIRECTION('', (0.0, -1.0, 0.0));
#4725=VECTOR('', #4724, 20.0);
#4726=LINE('', #1114, #4725);
#4727=EDGE_CURVE('', #1115, #4552, #4726, .T.);
#4728=ORIENTED_EDGE('', *, *, #4727, .T.);
#4729=ORIENTED_EDGE('', *, *, #4558, .T.);
#4730=DIRECTION('', (0.0, -1.0, 0.0));
#4731=VECTOR('', #4730, 20.0);
#4732=LINE('', #1116, #4731);
#4733=EDGE_CURVE('', #1117, #4554, #4732, .T.);
#4734=ORIENTED_EDGE('', *, *, #4733, .F.);

```

```

#4735=EDGE_LOOP('',( #4723, #4728, #4729, #4734));
#4736=FACE_BOUND('', #4735, .F.);
#4737=CARTESIAN_POINT('', (69.999999999999972, 0.0, 145.0));
#4738=DIRECTION('', (0.0, 0.0, 1.0));
#4739=DIRECTION('', (1.0, 0.0, 0.0));
#4740=AXIS2_PLACEMENT_3D('', #4737, #4738, #4739);
#4741=PLANE('', #4740);
#4742=ADVANCED_FACE('', (#4736), #4741, .F.);
#4743=SHAPE_ASPECT('featured shape: face 132', $, #38, .T.);
#4744=PROPERTY_DEFINITION('', $, #4743);
#4745=FACE_SHAPE_REPRESENTATION('', (#4742), #29);
#4746=PROPERTY_DEFINITION_REPRESENTATION(#4744, #4745);
#4747=ORIENTED_EDGE('', *, *, #1130, .T.);
#4748=ORIENTED_EDGE('', *, *, #4733, .T.);
#4749=ORIENTED_EDGE('', *, *, #4613, .T.);
#4750=DIRECTION('', (0.0, -1.0, 0.0));
#4751=VECTOR('', #4750, 20.0);
#4752=LINE('', #1123, #4751);
#4753=EDGE_CURVE('', #1124, #4602, #4752, .T.);
#4754=ORIENTED_EDGE('', *, *, #4753, .F.);
#4755=EDGE_LOOP('', (#4747, #4748, #4749, #4754));
#4756=FACE_BOUND('', #4755, .F.);
#4757=CARTESIAN_POINT('', (95.999999999999972, 0.0, 147.0));
#4758=DIRECTION('', (0.0, 1.0, 0.0));
#4759=DIRECTION('', (1.0, 0.0, 0.0));
#4760=AXIS2_PLACEMENT_3D('', #4757, #4758, #4759);
#4761=CYLINDRICAL_SURFACE('', #4760, 2.0);
#4762=ADVANCED_FACE('', (#4756), #4761, .T.);
#4763=SHAPE_ASPECT('featured shape: face 133', $, #38, .T.);
#4764=PROPERTY_DEFINITION('', $, #4763);
#4765=FACE_SHAPE_REPRESENTATION('', (#4762), #29);
#4766=PROPERTY_DEFINITION_REPRESENTATION(#4764, #4765);
#4767=ORIENTED_EDGE('', *, *, #1137, .F.);
#4768=ORIENTED_EDGE('', *, *, #4753, .T.);
#4769=ORIENTED_EDGE('', *, *, #4606, .T.);
#4770=DIRECTION('', (0.0, -1.0, 0.0));
#4771=VECTOR('', #4770, 20.0);
#4772=LINE('', #1132, #4771);
#4773=EDGE_CURVE('', #1133, #4593, #4772, .T.);
#4774=ORIENTED_EDGE('', *, *, #4773, .F.);
#4775=EDGE_LOOP('', (#4767, #4768, #4769, #4774));
#4776=FACE_BOUND('', #4775, .F.);
#4777=CARTESIAN_POINT('', (97.999999999999972, 0.0, 147.0));
#4778=DIRECTION('', (-1.0, 0.0, 0.0));
#4779=DIRECTION('', (0.0, 0.0, 1.0));
#4780=AXIS2_PLACEMENT_3D('', #4777, #4778, #4779);
#4781=PLANE('', #4780);
#4782=ADVANCED_FACE('', (#4776), #4781, .F.);
#4783=SHAPE_ASPECT('featured shape: face 134', $, #38, .T.);
#4784=PROPERTY_DEFINITION('', $, #4783);
#4785=FACE_SHAPE_REPRESENTATION('', (#4782), #29);
#4786=PROPERTY_DEFINITION_REPRESENTATION(#4784, #4785);
#4787=ORIENTED_EDGE('', *, *, #1146, .T.);

```

```

#4788=ORIENTED_EDGE('',*,*,#4773,.T.);
#4789=ORIENTED_EDGE('',*,*,#4599,.T.);
#4790=DIRECTION('',(0.0,-1.0,0.0));
#4791=VECTOR('',#4790,20.0);
#4792=LINE('',#1139,#4791);
#4793=EDGE_CURVE('',#1140,#4586,#4792,.T.);
#4794=ORIENTED_EDGE('',*,*,#4793,.F.);
#4795=EDGE_LOOP('',(#4787,#4788,#4789,#4794));
#4796=FACE_BOUND('',#4795,.F.);
#4797=CARTESIAN_POINT('',(95.999999999999972,0.0,163.0));
#4798=DIRECTION('',(0.0,1.0,0.0));
#4799=DIRECTION('',(1.0,0.0,0.0));
#4800=AXIS2_PLACEMENT_3D('',#4797,#4798,#4799);
#4801=CYLINDRICAL_SURFACE('',#4800,2.0);
#4802=ADVANCED_FACE('',(#4796),#4801,.T.);
#4803=SHAPE_ASPECT('featured shape: face 135',$,#38,.T.);
#4804=PROPERTY_DEFINITION('',$,#4803);
#4805=FACE_SHAPE_REPRESENTATION('',(#4802),#29);
#4806=PROPERTY_DEFINITION_REPRESENTATION(#4804,#4805);
#4807=ORIENTED_EDGE('',*,*,#1153,.F.);
#4808=ORIENTED_EDGE('',*,*,#4793,.T.);
#4809=ORIENTED_EDGE('',*,*,#4590,.T.);
#4810=DIRECTION('',(0.0,-1.0,0.0));
#4811=VECTOR('',#4810,20.0);
#4812=LINE('',#1148,#4811);
#4813=EDGE_CURVE('',#1149,#4577,#4812,.T.);
#4814=ORIENTED_EDGE('',*,*,#4813,.F.);
#4815=EDGE_LOOP('',(#4807,#4808,#4809,#4814));
#4816=FACE_BOUND('',#4815,.F.);
#4817=CARTESIAN_POINT('',(95.999999999999972,0.0,165.0));
#4818=DIRECTION('',(0.0,0.0,-1.0));
#4819=DIRECTION('',(-1.0,0.0,0.0));
#4820=AXIS2_PLACEMENT_3D('',#4817,#4818,#4819);
#4821=PLANE('',#4820);
#4822=ADVANCED_FACE('',(#4816),#4821,.F.);
#4823=SHAPE_ASPECT('featured shape: face 136',$,#38,.T.);
#4824=PROPERTY_DEFINITION('',$,#4823);
#4825=FACE_SHAPE_REPRESENTATION('',(#4822),#29);
#4826=PROPERTY_DEFINITION_REPRESENTATION(#4824,#4825);
#4827=ORIENTED_EDGE('',*,*,#1162,.T.);
#4828=ORIENTED_EDGE('',*,*,#4813,.T.);
#4829=ORIENTED_EDGE('',*,*,#4583,.T.);
#4830=DIRECTION('',(0.0,-1.0,0.0));
#4831=VECTOR('',#4830,20.0);
#4832=LINE('',#1155,#4831);
#4833=EDGE_CURVE('',#1156,#4570,#4832,.T.);
#4834=ORIENTED_EDGE('',*,*,#4833,.F.);
#4835=EDGE_LOOP('',(#4827,#4828,#4829,#4834));
#4836=FACE_BOUND('',#4835,.F.);
#4837=CARTESIAN_POINT('',(69.999999999999972,0.0,163.0));
#4838=DIRECTION('',(0.0,1.0,0.0));
#4839=DIRECTION('',(1.0,0.0,0.0));
#4840=AXIS2_PLACEMENT_3D('',#4837,#4838,#4839);

```

```

#4841=CYLINDRICAL_SURFACE('',#4840,2.0);
#4842=ADVANCED_FACE('',(#4836),#4841,.T.);
#4843=SHAPE_ASPECT('featured shape: face 137',$,#38,.T.);
#4844=PROPERTY_DEFINITION('',$,#4843);
#4845=FACE_SHAPE_REPRESENTATION('',(#4842),#29);
#4846=PROPERTY_DEFINITION_REPRESENTATION(#4844,#4845);
#4847=ORIENTED_EDGE('',*,*,#1169,.F.);
#4848=ORIENTED_EDGE('',*,*,#4833,.T.);
#4849=ORIENTED_EDGE('',*,*,#4574,.T.);
#4850=DIRECTION('',(0.0,-1.0,0.0));
#4851=VECTOR('',#4850,20.0);
#4852=LINE('',#1164,#4851);
#4853=EDGE_CURVE('',#1165,#4561,#4852,.T.);
#4854=ORIENTED_EDGE('',*,*,#4853,.F.);
#4855=EDGE_LOOP('',(#4847,#4848,#4849,#4854));
#4856=FACE_BOUND('',#4855,.F.);
#4857=CARTESIAN_POINT('',(67.999999999999972,0.0,163.0));
#4858=DIRECTION('',(1.0,0.0,0.0));
#4859=DIRECTION('',(0.0,0.0,-1.0));
#4860=AXIS2_PLACEMENT_3D('',#4857,#4858,#4859);
#4861=PLANE('',#4860);
#4862=ADVANCED_FACE('',(#4856),#4861,.F.);
#4863=SHAPE_ASPECT('featured shape: face 138',$,#38,.T.);
#4864=PROPERTY_DEFINITION('',$,#4863);
#4865=FACE_SHAPE_REPRESENTATION('',(#4862),#29);
#4866=PROPERTY_DEFINITION_REPRESENTATION(#4864,#4865);
#4867=ORIENTED_EDGE('',*,*,#1176,.T.);
#4868=ORIENTED_EDGE('',*,*,#4853,.T.);
#4869=ORIENTED_EDGE('',*,*,#4567,.T.);
#4870=ORIENTED_EDGE('',*,*,#4727,.F.);
#4871=EDGE_LOOP('',(#4867,#4868,#4869,#4870));
#4872=FACE_BOUND('',#4871,.F.);
#4873=CARTESIAN_POINT('',(69.999999999999972,0.0,147.0));
#4874=DIRECTION('',(0.0,1.0,0.0));
#4875=DIRECTION('',(1.0,0.0,0.0));
#4876=AXIS2_PLACEMENT_3D('',#4873,#4874,#4875);
#4877=CYLINDRICAL_SURFACE('',#4876,2.0);
#4878=ADVANCED_FACE('',(#4872),#4877,.T.);
#4879=SHAPE_ASPECT('featured shape: face 139',$,#38,.T.);
#4880=PROPERTY_DEFINITION('',$,#4879);
#4881=FACE_SHAPE_REPRESENTATION('',(#4878),#29);
#4882=PROPERTY_DEFINITION_REPRESENTATION(#4880,#4881);
#4883=ORIENTED_EDGE('',*,*,#250,.F.);
#4884=DIRECTION('',(0.0,1.0,0.0));
#4885=VECTOR('',#4884,10.0);
#4886=LINE('',#243,#4885);
#4887=EDGE_CURVE('',#244,#1482,#4886,.T.);
#4888=ORIENTED_EDGE('',*,*,#4887,.T.);
#4889=ORIENTED_EDGE('',*,*,#1486,.T.);
#4890=DIRECTION('',(0.0,1.0,0.0));
#4891=VECTOR('',#4890,10.0);
#4892=LINE('',#245,#4891);
#4893=EDGE_CURVE('',#246,#1473,#4892,.T.);

```

```

#4894=ORIENTED_EDGE('',*,*,#4893,.F.);
#4895=EDGE_LOOP('',(#4883,#4888,#4889,#4894));
#4896=FACE_BOUND('',#4895,.F.);
#4897=CARTESIAN_POINT('',(2.0,-50.0,50.000000000000007));
#4898=DIRECTION('',(0.0,0.0,-1.0));
#4899=DIRECTION('',(1.0,0.0,0.0));
#4900=AXIS2_PLACEMENT_3D('',#4897,#4898,#4899);
#4901=PLANE('',#4900);
#4902=ADVANCED_FACE('',(#4896),#4901,.F.);
#4903=SHAPE_ASPECT('featured shape: face 140',$,#38,.T.);
#4904=PROPERTY_DEFINITION('',$,#4903);
#4905=FACE_SHAPE_REPRESENTATION('',(#4902),#29);
#4906=PROPERTY_DEFINITION_REPRESENTATION(#4904,#4905);
#4907=ORIENTED_EDGE('',*,*,#259,.T.);
#4908=ORIENTED_EDGE('',*,*,#4893,.T.);
#4909=ORIENTED_EDGE('',*,*,#1479,.T.);
#4910=DIRECTION('',(0.0,1.0,0.0));
#4911=VECTOR('',#4910,10.0);
#4912=LINE('',#252,#4911);
#4913=EDGE_CURVE('',#253,#1466,#4912,.T.);
#4914=ORIENTED_EDGE('',*,*,#4913,.F.);
#4915=EDGE_LOOP('',(#4907,#4908,#4909,#4914));
#4916=FACE_BOUND('',#4915,.F.);
#4917=CARTESIAN_POINT('',(45.0,-50.0,45.000000000000007));
#4918=DIRECTION('',(0.0,-1.0,0.0));
#4919=DIRECTION('',(1.0,0.0,0.0));
#4920=AXIS2_PLACEMENT_3D('',#4917,#4918,#4919);
#4921=CYLINDRICAL_SURFACE('',#4920,5.0);
#4922=ADVANCED_FACE('',(#4916),#4921,.T.);
#4923=SHAPE_ASPECT('featured shape: face 141',$,#38,.T.);
#4924=PROPERTY_DEFINITION('',$,#4923);
#4925=FACE_SHAPE_REPRESENTATION('',(#4922),#29);
#4926=PROPERTY_DEFINITION_REPRESENTATION(#4924,#4925);
#4927=ORIENTED_EDGE('',*,*,#266,.T.);
#4928=ORIENTED_EDGE('',*,*,#4913,.T.);
#4929=ORIENTED_EDGE('',*,*,#1470,.T.);
#4930=DIRECTION('',(0.0,1.0,0.0));
#4931=VECTOR('',#4930,10.0);
#4932=LINE('',#261,#4931);
#4933=EDGE_CURVE('',#262,#1457,#4932,.T.);
#4934=ORIENTED_EDGE('',*,*,#4933,.F.);
#4935=EDGE_LOOP('',(#4927,#4928,#4929,#4934));
#4936=FACE_BOUND('',#4935,.F.);
#4937=CARTESIAN_POINT('',(50.0,-50.0,45.000000000000007));
#4938=DIRECTION('',(-1.0,0.0,0.0));
#4939=DIRECTION('',(0.0,0.0,-1.0));
#4940=AXIS2_PLACEMENT_3D('',#4937,#4938,#4939);
#4941=PLANE('',#4940);
#4942=ADVANCED_FACE('',(#4936),#4941,.F.);
#4943=SHAPE_ASPECT('featured shape: face 142',$,#38,.T.);
#4944=PROPERTY_DEFINITION('',$,#4943);
#4945=FACE_SHAPE_REPRESENTATION('',(#4942),#29);
#4946=PROPERTY_DEFINITION_REPRESENTATION(#4944,#4945);

```

```

#4947=ORIENTED_EDGE('',*,*,#346,.F.);
#4948=DIRECTION('',(0.0,1.0,0.0));
#4949=VECTOR('',#4948,10.0);
#4950=LINE('',#339,#4949);
#4951=EDGE_CURVE('',#340,#1548,#4950,.T.);
#4952=ORIENTED_EDGE('',*,*,#4951,.T.);
#4953=ORIENTED_EDGE('',*,*,#1552,.T.);
#4954=DIRECTION('',(0.0,1.0,0.0));
#4955=VECTOR('',#4954,10.0);
#4956=LINE('',#341,#4955);
#4957=EDGE_CURVE('',#342,#1539,#4956,.T.);
#4958=ORIENTED_EDGE('',*,*,#4957,.F.);
#4959=EDGE_LOOP('',(#4947,#4952,#4953,#4958));
#4960=FACE_BOUND('',#4959,.F.);
#4961=CARTESIAN_POINT('',(180.000000000000028,-50.0,2.0));
#4962=DIRECTION('',(1.0,0.0,0.0));
#4963=DIRECTION('',(0.0,0.0,1.0));
#4964=AXIS2_PLACEMENT_3D('',#4961,#4962,#4963);
#4965=PLANE('',#4964);
#4966=ADVANCED_FACE('',(#4960),#4965,.F.);
#4967=SHAPE_ASPECT('featured shape: face 143',$,#38,.T.);
#4968=PROPERTY_DEFINITION('',$,#4967);
#4969=FACE_SHAPE_REPRESENTATION('',(#4966),#29);
#4970=PROPERTY_DEFINITION_REPRESENTATION(#4968,#4969);
#4971=ORIENTED_EDGE('',*,*,#355,.T.);
#4972=ORIENTED_EDGE('',*,*,#4957,.T.);
#4973=ORIENTED_EDGE('',*,*,#1545,.T.);
#4974=DIRECTION('',(0.0,1.0,0.0));
#4975=VECTOR('',#4974,10.0);
#4976=LINE('',#348,#4975);
#4977=EDGE_CURVE('',#349,#1532,#4976,.T.);
#4978=ORIENTED_EDGE('',*,*,#4977,.F.);
#4979=EDGE_LOOP('',(#4971,#4972,#4973,#4978));
#4980=FACE_BOUND('',#4979,.F.);
#4981=CARTESIAN_POINT('',(185.000000000000028,-50.0,45.0));
#4982=DIRECTION('',(0.0,-1.0,0.0));
#4983=DIRECTION('',(1.0,0.0,0.0));
#4984=AXIS2_PLACEMENT_3D('',#4981,#4982,#4983);
#4985=CYLINDRICAL_SURFACE('',#4984,5.0);
#4986=ADVANCED_FACE('',(#4980),#4985,.T.);
#4987=SHAPE_ASPECT('featured shape: face 144',$,#38,.T.);
#4988=PROPERTY_DEFINITION('',$,#4987);
#4989=FACE_SHAPE_REPRESENTATION('',(#4986),#29);
#4990=PROPERTY_DEFINITION_REPRESENTATION(#4988,#4989);
#4991=ORIENTED_EDGE('',*,*,#362,.T.);
#4992=ORIENTED_EDGE('',*,*,#4977,.T.);
#4993=ORIENTED_EDGE('',*,*,#1536,.T.);
#4994=DIRECTION('',(0.0,1.0,0.0));
#4995=VECTOR('',#4994,10.0);
#4996=LINE('',#357,#4995);
#4997=EDGE_CURVE('',#358,#1523,#4996,.T.);
#4998=ORIENTED_EDGE('',*,*,#4997,.F.);
#4999=EDGE_LOOP('',(#4991,#4992,#4993,#4998));

```

```

#5000=FACE_BOUND('',#4999,.F.);
#5001=CARTESIAN_POINT('',(185.0000000000000028,-50.0,50.0));
#5002=DIRECTION('',(0.0,0.0,-1.0));
#5003=DIRECTION('',(1.0,0.0,0.0));
#5004=AXIS2_PLACEMENT_3D('',#5001,#5002,#5003);
#5005=PLANE('',#5004);
#5006=ADVANCED_FACE('',(#5000),#5005,.F.);
#5007=SHAPE_ASPECT('featured shape: face 145',$,#38,.T.);
#5008=PROPERTY_DEFINITION('',$,#5007);
#5009=FACE_SHAPE_REPRESENTATION('',(#5006),#29);
#5010=PROPERTY_DEFINITION_REPRESENTATION(#5008,#5009);
#5011=ORIENTED_EDGE('',*,*,#437,.F.);
#5012=DIRECTION('',(0.0,1.0,0.0));
#5013=VECTOR('',#5012,10.0);
#5014=LINE('',#430,#5013);
#5015=EDGE_CURVE('',#431,#1614,#5014,.T.);
#5016=ORIENTED_EDGE('',*,*,#5015,.T.);
#5017=ORIENTED_EDGE('',*,*,#1618,.T.);
#5018=DIRECTION('',(0.0,1.0,0.0));
#5019=VECTOR('',#5018,10.0);
#5020=LINE('',#432,#5019);
#5021=EDGE_CURVE('',#433,#1605,#5020,.T.);
#5022=ORIENTED_EDGE('',*,*,#5021,.F.);
#5023=EDGE_LOOP('',(#5011,#5016,#5017,#5022));
#5024=FACE_BOUND('',#5023,.F.);
#5025=CARTESIAN_POINT('',(228.0,-50.0,130.0000000000000028));
#5026=DIRECTION('',(0.0,0.0,1.0));
#5027=DIRECTION('',(-1.0,0.0,0.0));
#5028=AXIS2_PLACEMENT_3D('',#5025,#5026,#5027);
#5029=PLANE('',#5028);
#5030=ADVANCED_FACE('',(#5024),#5029,.F.);
#5031=SHAPE_ASPECT('featured shape: face 146',$,#38,.T.);
#5032=PROPERTY_DEFINITION('',$,#5031);
#5033=FACE_SHAPE_REPRESENTATION('',(#5030),#29);
#5034=PROPERTY_DEFINITION_REPRESENTATION(#5032,#5033);
#5035=ORIENTED_EDGE('',*,*,#446,.T.);
#5036=ORIENTED_EDGE('',*,*,#5021,.T.);
#5037=ORIENTED_EDGE('',*,*,#1611,.T.);
#5038=DIRECTION('',(0.0,1.0,0.0));
#5039=VECTOR('',#5038,10.0);
#5040=LINE('',#439,#5039);
#5041=EDGE_CURVE('',#440,#1598,#5040,.T.);
#5042=ORIENTED_EDGE('',*,*,#5041,.F.);
#5043=EDGE_LOOP('',(#5035,#5036,#5037,#5042));
#5044=FACE_BOUND('',#5043,.F.);
#5045=CARTESIAN_POINT('',(184.9999999999999972,-50.0,135.0000000000000028));
#5046=DIRECTION('',(0.0,-1.0,0.0));
#5047=DIRECTION('',(1.0,0.0,0.0));
#5048=AXIS2_PLACEMENT_3D('',#5045,#5046,#5047);
#5049=CYLINDRICAL_SURFACE('',#5048,5.0);
#5050=ADVANCED_FACE('',(#5044),#5049,.T.);
#5051=SHAPE_ASPECT('featured shape: face 147',$,#38,.T.);
#5052=PROPERTY_DEFINITION('',$,#5051);

```

```

#5053=FACE_SHAPE_REPRESENTATION('', (#5050), #29);
#5054=PROPERTY_DEFINITION_REPRESENTATION(#5052, #5053);
#5055=ORIENTED_EDGE('', *, *, #453, .T.);
#5056=ORIENTED_EDGE('', *, *, #5041, .T.);
#5057=ORIENTED_EDGE('', *, *, #1602, .T.);
#5058=DIRECTION('', (0.0, 1.0, 0.0));
#5059=VECTOR('', #5058, 10.0);
#5060=LINE('', #448, #5059);
#5061=EDGE_CURVE('', #449, #1589, #5060, .T.);
#5062=ORIENTED_EDGE('', *, *, #5061, .F.);
#5063=EDGE_LOOP('', (#5055, #5056, #5057, #5062));
#5064=FACE_BOUND('', #5063, .F.);
#5065=CARTESIAN_POINT('', (179.999999999999972, -50.0, 135.000000000000028));
#5066=DIRECTION('', (1.0, 0.0, 0.0));
#5067=DIRECTION('', (0.0, 0.0, 1.0));
#5068=AXIS2_PLACEMENT_3D('', #5065, #5066, #5067);
#5069=PLANE('', #5068);
#5070=ADVANCED_FACE('', (#5064), #5069, .F.);
#5071=SHAPE_ASPECT('featured shape: face 148', $, #38, .T.);
#5072=PROPERTY_DEFINITION('', $, #5071);
#5073=FACE_SHAPE_REPRESENTATION('', (#5070), #29);
#5074=PROPERTY_DEFINITION_REPRESENTATION(#5072, #5073);
#5075=ORIENTED_EDGE('', *, *, #528, .F.);
#5076=DIRECTION('', (0.0, 1.0, 0.0));
#5077=VECTOR('', #5076, 10.0);
#5078=LINE('', #521, #5077);
#5079=EDGE_CURVE('', #522, #1760, #5078, .T.);
#5080=ORIENTED_EDGE('', *, *, #5079, .T.);
#5081=ORIENTED_EDGE('', *, *, #1764, .T.);
#5082=DIRECTION('', (0.0, 1.0, 0.0));
#5083=VECTOR('', #5082, 10.0);
#5084=LINE('', #523, #5083);
#5085=EDGE_CURVE('', #524, #1751, #5084, .T.);
#5086=ORIENTED_EDGE('', *, *, #5085, .F.);
#5087=EDGE_LOOP('', (#5075, #5080, #5081, #5086));
#5088=FACE_BOUND('', #5087, .F.);
#5089=CARTESIAN_POINT('', (49.999999999999972, -50.0, 178.0));
#5090=DIRECTION('', (-1.0, 0.0, 0.0));
#5091=DIRECTION('', (0.0, 0.0, -1.0));
#5092=AXIS2_PLACEMENT_3D('', #5089, #5090, #5091);
#5093=PLANE('', #5092);
#5094=ADVANCED_FACE('', (#5088), #5093, .F.);
#5095=SHAPE_ASPECT('featured shape: face 149', $, #38, .T.);
#5096=PROPERTY_DEFINITION('', $, #5095);
#5097=FACE_SHAPE_REPRESENTATION('', (#5094), #29);
#5098=PROPERTY_DEFINITION_REPRESENTATION(#5096, #5097);
#5099=ORIENTED_EDGE('', *, *, #537, .T.);
#5100=ORIENTED_EDGE('', *, *, #5085, .T.);
#5101=ORIENTED_EDGE('', *, *, #1757, .T.);
#5102=DIRECTION('', (0.0, 1.0, 0.0));
#5103=VECTOR('', #5102, 10.0);
#5104=LINE('', #530, #5103);
#5105=EDGE_CURVE('', #531, #1744, #5104, .T.);

```



```

#5106=ORIENTED_EDGE('',*,*,#5105,.F.);
#5107=EDGE_LOOP('',(#5099,#5100,#5101,#5106));
#5108=FACE_BOUND('',#5107,.F.);
#5109=CARTESIAN_POINT('',(44.999999999999972,-50.0,134.99999999999972));
#5110=DIRECTION('',(0.0,-1.0,0.0));
#5111=DIRECTION('',(1.0,0.0,0.0));
#5112=AXIS2_PLACEMENT_3D('',#5109,#5110,#5111);
#5113=CYLINDRICAL_SURFACE('',#5112,5.0);
#5114=ADVANCED_FACE('',(#5108),#5113,.T.);
#5115=SHAPE_ASPECT('featured shape: face 150',$,#38,.T.);
#5116=PROPERTY_DEFINITION('',$,#5115);
#5117=FACE_SHAPE_REPRESENTATION('',(#5114),#29);
#5118=PROPERTY_DEFINITION_REPRESENTATION(#5116,#5117);
#5119=ORIENTED_EDGE('',*,*,#544,.T.);
#5120=ORIENTED_EDGE('',*,*,#5105,.T.);
#5121=ORIENTED_EDGE('',*,*,#1748,.T.);
#5122=DIRECTION('',(0.0,1.0,0.0));
#5123=VECTOR('',#5122,10.0);
#5124=LINE('',#539,#5123);
#5125=EDGE_CURVE('',#540,#1735,#5124,.T.);
#5126=ORIENTED_EDGE('',*,*,#5125,.F.);
#5127=EDGE_LOOP('',(#5119,#5120,#5121,#5126));
#5128=FACE_BOUND('',#5127,.F.);
#5129=CARTESIAN_POINT('',(44.999999999999972,-50.0,129.99999999999972));
#5130=DIRECTION('',(0.0,0.0,1.0));
#5131=DIRECTION('',(-1.0,0.0,0.0));
#5132=AXIS2_PLACEMENT_3D('',#5129,#5130,#5131);
#5133=PLANE('',#5132);
#5134=ADVANCED_FACE('',(#5128),#5133,.F.);
#5135=SHAPE_ASPECT('featured shape: face 151',$,#38,.T.);
#5136=PROPERTY_DEFINITION('',$,#5135);
#5137=FACE_SHAPE_REPRESENTATION('',(#5134),#29);
#5138=PROPERTY_DEFINITION_REPRESENTATION(#5136,#5137);
#5139=ORIENTED_EDGE('',*,*,#2271,.T.);
#5140=ORIENTED_EDGE('',*,*,#1561,.T.);
#5141=ORIENTED_EDGE('',*,*,#4951,.F.);
#5142=ORIENTED_EDGE('',*,*,#401,.F.);
#5143=EDGE_LOOP('',(#5139,#5140,#5141,#5142));
#5144=FACE_BOUND('',#5143,.F.);
#5145=CARTESIAN_POINT('',(185.000000000000028,-50.0,7.0));
#5146=DIRECTION('',(0.0,1.0,0.0));
#5147=DIRECTION('',(0.0,0.0,-1.0));
#5148=AXIS2_PLACEMENT_3D('',#5145,#5146,#5147);
#5149=CYLINDRICAL_SURFACE('',#5148,5.0);
#5150=ADVANCED_FACE('',(#5144),#5149,.T.);
#5151=SHAPE_ASPECT('featured shape: face 152',$,#38,.T.);
#5152=PROPERTY_DEFINITION('',$,#5151);
#5153=FACE_SHAPE_REPRESENTATION('',(#5150),#29);
#5154=PROPERTY_DEFINITION_REPRESENTATION(#5152,#5153);
#5155=ORIENTED_EDGE('',*,*,#2204,.F.);
#5156=ORIENTED_EDGE('',*,*,#462,.T.);
#5157=ORIENTED_EDGE('',*,*,#5061,.T.);
#5158=ORIENTED_EDGE('',*,*,#1595,.F.);

```

```

#5159=EDGE_LOOP('', (#5155, #5156, #5157, #5158));
#5160=FACE_BOUND('', #5159, .F.);
#5161=CARTESIAN_POINT('', (184.999999999999972, -50.0, 173.0));
#5162=DIRECTION('', (0.0, -1.0, 0.0));
#5163=DIRECTION('', (0.0, 0.0, 1.0));
#5164=AXIS2_PLACEMENT_3D('', #5161, #5162, #5163);
#5165=CYLINDRICAL_SURFACE('', #5164, 5.0);
#5166=ADVANCED_FACE('', (#5160), #5165, .T.);
#5167=SHAPE_ASPECT('featured shape: face 153', $, #38, .T.);
#5168=PROPERTY_DEFINITION('', $, #5167);
#5169=FACE_SHAPE_REPRESENTATION('', (#5166), #29);
#5170=PROPERTY_DEFINITION_REPRESENTATION(#5168, #5169);
#5171=ORIENTED_EDGE('', *, *, #2161, .T.);
#5172=ORIENTED_EDGE('', *, *, #1495, .T.);
#5173=ORIENTED_EDGE('', *, *, #4887, .F.);
#5174=ORIENTED_EDGE('', *, *, #305, .F.);
#5175=EDGE_LOOP('', (#5171, #5172, #5173, #5174));
#5176=FACE_BOUND('', #5175, .F.);
#5177=CARTESIAN_POINT('', (7.0, -50.0, 45.000000000000007));
#5178=DIRECTION('', (0.0, 1.0, 0.0));
#5179=DIRECTION('', (-1.0, 0.0, 0.0));
#5180=AXIS2_PLACEMENT_3D('', #5177, #5178, #5179);
#5181=CYLINDRICAL_SURFACE('', #5180, 5.0);
#5182=ADVANCED_FACE('', (#5176), #5181, .T.);
#5183=SHAPE_ASPECT('featured shape: face 154', $, #38, .T.);
#5184=PROPERTY_DEFINITION('', $, #5183);
#5185=FACE_SHAPE_REPRESENTATION('', (#5182), #29);
#5186=PROPERTY_DEFINITION_REPRESENTATION(#5184, #5185);
#5187=ORIENTED_EDGE('', *, *, #2118, .F.);
#5188=ORIENTED_EDGE('', *, *, #371, .T.);
#5189=ORIENTED_EDGE('', *, *, #4997, .T.);
#5190=ORIENTED_EDGE('', *, *, #1529, .F.);
#5191=EDGE_LOOP('', (#5187, #5188, #5189, #5190));
#5192=FACE_BOUND('', #5191, .F.);
#5193=CARTESIAN_POINT('', (223.0, -50.0, 45.0));
#5194=DIRECTION('', (0.0, -1.0, 0.0));
#5195=DIRECTION('', (1.0, 0.0, 0.0));
#5196=AXIS2_PLACEMENT_3D('', #5193, #5194, #5195);
#5197=CYLINDRICAL_SURFACE('', #5196, 5.0);
#5198=ADVANCED_FACE('', (#5192), #5197, .T.);
#5199=SHAPE_ASPECT('featured shape: face 155', $, #38, .T.);
#5200=PROPERTY_DEFINITION('', $, #5199);
#5201=FACE_SHAPE_REPRESENTATION('', (#5198), #29);
#5202=PROPERTY_DEFINITION_REPRESENTATION(#5200, #5201);
#5203=ORIENTED_EDGE('', *, *, #2142, .T.);
#5204=ORIENTED_EDGE('', *, *, #1627, .T.);
#5205=ORIENTED_EDGE('', *, *, #5015, .F.);
#5206=ORIENTED_EDGE('', *, *, #492, .F.);
#5207=EDGE_LOOP('', (#5203, #5204, #5205, #5206));
#5208=FACE_BOUND('', #5207, .F.);
#5209=CARTESIAN_POINT('', (223.0, -50.0, 135.000000000000028));
#5210=DIRECTION('', (0.0, 1.0, 0.0));
#5211=DIRECTION('', (1.0, 0.0, 0.0));

```

```

#5212=AXIS2_PLACEMENT_3D('',#5209,#5210,#5211);
#5213=CYLINDRICAL_SURFACE('',#5212,5.0);
#5214=ADVANCED_FACE('',(#5208),#5213,.T.);
#5215=SHAPE_ASPECT('featured shape: face 156',$,#38,.T.);
#5216=PROPERTY_DEFINITION('',$,#5215);
#5217=FACE_SHAPE_REPRESENTATION('',(#5214),#29);
#5218=PROPERTY_DEFINITION_REPRESENTATION(#5216,#5217);
#5219=ORIENTED_EDGE('',*,*,#2247,.F.);
#5220=ORIENTED_EDGE('',*,*,#275,.T.);
#5221=ORIENTED_EDGE('',*,*,#4933,.T.);
#5222=ORIENTED_EDGE('',*,*,#1463,.F.);
#5223=EDGE_LOOP('',(#5219,#5220,#5221,#5222));
#5224=FACE_BOUND('',#5223,.F.);
#5225=CARTESIAN_POINT('',(45.0,-50.0,7.0));
#5226=DIRECTION('',(0.0,-1.0,0.0));
#5227=DIRECTION('',(0.0,0.0,-1.0));
#5228=AXIS2_PLACEMENT_3D('',#5225,#5226,#5227);
#5229=CYLINDRICAL_SURFACE('',#5228,5.0);
#5230=ADVANCED_FACE('',(#5224),#5229,.T.);
#5231=SHAPE_ASPECT('featured shape: face 157',$,#38,.T.);
#5232=PROPERTY_DEFINITION('',$,#5231);
#5233=FACE_SHAPE_REPRESENTATION('',(#5230),#29);
#5234=PROPERTY_DEFINITION_REPRESENTATION(#5232,#5233);
#5235=ORIENTED_EDGE('',*,*,#2228,.T.);
#5236=ORIENTED_EDGE('',*,*,#1773,.T.);
#5237=ORIENTED_EDGE('',*,*,#5079,.F.);
#5238=ORIENTED_EDGE('',*,*,#583,.F.);
#5239=EDGE_LOOP('',(#5235,#5236,#5237,#5238));
#5240=FACE_BOUND('',#5239,.F.);
#5241=CARTESIAN_POINT('',(44.999999999999972,-50.0,173.0));
#5242=DIRECTION('',(0.0,1.0,0.0));
#5243=DIRECTION('',(0.0,0.0,1.0));
#5244=AXIS2_PLACEMENT_3D('',#5241,#5242,#5243);
#5245=CYLINDRICAL_SURFACE('',#5244,5.0);
#5246=ADVANCED_FACE('',(#5240),#5245,.T.);
#5247=SHAPE_ASPECT('featured shape: face 158',$,#38,.T.);
#5248=PROPERTY_DEFINITION('',$,#5247);
#5249=FACE_SHAPE_REPRESENTATION('',(#5246),#29);
#5250=PROPERTY_DEFINITION_REPRESENTATION(#5248,#5249);
#5251=ORIENTED_EDGE('',*,*,#2185,.F.);
#5252=ORIENTED_EDGE('',*,*,#553,.T.);
#5253=ORIENTED_EDGE('',*,*,#5125,.T.);
#5254=ORIENTED_EDGE('',*,*,#1741,.F.);
#5255=EDGE_LOOP('',(#5251,#5252,#5253,#5254));
#5256=FACE_BOUND('',#5255,.F.);
#5257=CARTESIAN_POINT('',(7.0,-50.0,134.999999999999972));
#5258=DIRECTION('',(0.0,-1.0,0.0));
#5259=DIRECTION('',(-1.0,0.0,0.0));
#5260=AXIS2_PLACEMENT_3D('',#5257,#5258,#5259);
#5261=CYLINDRICAL_SURFACE('',#5260,5.0);
#5262=ADVANCED_FACE('',(#5256),#5261,.T.);
#5263=SHAPE_ASPECT('featured shape: face 159',$,#38,.T.);
#5264=PROPERTY_DEFINITION('',$,#5263);

```

```

#5265=FACE_SHAPE_REPRESENTATION('', (#5262), #29);
#5266=PROPERTY_DEFINITION_REPRESENTATION(#5264, #5265);
#5267=ORIENTED_EDGE('', *, *, #217, .T.);
#5268=ORIENTED_EDGE('', *, *, #2094, .F.);
#5269=ORIENTED_EDGE('', *, *, #4494, .T.);
#5270=ORIENTED_EDGE('', *, *, #2068, .F.);
#5271=ORIENTED_EDGE('', *, *, #161, .T.);
#5272=CARTESIAN_POINT('', (50.099999999999994, -10.0, 106.0));
#5273=VERTEX_POINT('', #5272);
#5274=DIRECTION('', (1.0, 0.0, 0.0));
#5275=VECTOR('', #5274, 50.099999999999994);
#5276=LINE('', #149, #5275);
#5277=EDGE_CURVE('', #150, #5273, #5276, .T.);
#5278=ORIENTED_EDGE('', *, *, #5277, .T.);
#5279=CARTESIAN_POINT('', (66.0, -10.0, 90.099999999999994));
#5280=VERTEX_POINT('', #5279);
#5281=CARTESIAN_POINT('', (50.099999999999994, -10.0, 90.099999999999994));
#5282=DIRECTION('', (0.0, 1.0, 0.0));
#5283=DIRECTION('', (0.0, 0.0, 1.0));
#5284=AXIS2_PLACEMENT_3D('', #5281, #5282, #5283);
#5285=CIRCLE('', #5284, 15.900000000000000);
#5286=EDGE_CURVE('', #5273, #5280, #5285, .T.);
#5287=ORIENTED_EDGE('', *, *, #5286, .T.);
#5288=CARTESIAN_POINT('', (66.0, -10.0, 89.900000000000006));
#5289=VERTEX_POINT('', #5288);
#5290=DIRECTION('', (0.0, 0.0, -1.0));
#5291=VECTOR('', #5290, 0.1999999999999989);
#5292=LINE('', #5279, #5291);
#5293=EDGE_CURVE('', #5280, #5289, #5292, .T.);
#5294=ORIENTED_EDGE('', *, *, #5293, .T.);
#5295=CARTESIAN_POINT('', (50.099999999999994, -10.0, 74.0));
#5296=VERTEX_POINT('', #5295);
#5297=CARTESIAN_POINT('', (50.099999999999994, -10.0, 89.900000000000006));
#5298=DIRECTION('', (0.0, 1.0, 0.0));
#5299=DIRECTION('', (1.0, 0.0, 0.0));
#5300=AXIS2_PLACEMENT_3D('', #5297, #5298, #5299);
#5301=CIRCLE('', #5300, 15.900000000000000);
#5302=EDGE_CURVE('', #5289, #5296, #5301, .T.);
#5303=ORIENTED_EDGE('', *, *, #5302, .T.);
#5304=DIRECTION('', (-1.0, 0.0, 0.0));
#5305=VECTOR('', #5304, 50.099999999999994);
#5306=LINE('', #5295, #5305);
#5307=EDGE_CURVE('', #5296, #213, #5306, .T.);
#5308=ORIENTED_EDGE('', *, *, #5307, .T.);
#5309=EDGE_LOOP('', (#5267, #5268, #5269, #5270, #5271, #5278, #5287, #5294, #5303, #5308));
#5310=FACE_BOUND('', #5309, .F.);
#5311=CARTESIAN_POINT('', (260.0, -10.0, 0.0));
#5312=DIRECTION('', (0.0, 1.0, 0.0));
#5313=DIRECTION('', (0.0, 0.0, 1.0));
#5314=AXIS2_PLACEMENT_3D('', #5311, #5312, #5313);
#5315=PLANE('', #5314);
#5316=ADVANCED_FACE('', (#5310), #5315, .F.);

```

```

#5317=SHAPE_ASPECT('featured shape: face 160', $, #38, .T.);
#5318=PROPERTY_DEFINITION('', $, #5317);
#5319=FACE_SHAPE_REPRESENTATION('', (#5316), #29);
#5320=PROPERTY_DEFINITION_REPRESENTATION(#5318, #5319);
#5321=ORIENTED_EDGE('', *, *, #229, .F.);
#5322=CARTESIAN_POINT('', (50.099999999999994, -30.0, 74.0));
#5323=VERTEX_POINT('', #5322);
#5324=DIRECTION('', (-1.0, 0.0, 0.0));
#5325=VECTOR('', #5324, 50.099999999999994);
#5326=LINE('', #5322, #5325);
#5327=EDGE_CURVE('', #5323, #220, #5326, .T.);
#5328=ORIENTED_EDGE('', *, *, #5327, .F.);
#5329=CARTESIAN_POINT('', (66.0, -30.0, 89.900000000000006));
#5330=VERTEX_POINT('', #5329);
#5331=CARTESIAN_POINT('', (50.099999999999994, -30.0, 89.900000000000006));
#5332=DIRECTION('', (0.0, 1.0, 0.0));
#5333=DIRECTION('', (1.0, 0.0, 0.0));
#5334=AXIS2_PLACEMENT_3D('', #5331, #5332, #5333);
#5335=CIRCLE('', #5334, 15.900000000000000);
#5336=EDGE_CURVE('', #5330, #5323, #5335, .T.);
#5337=ORIENTED_EDGE('', *, *, #5336, .F.);
#5338=CARTESIAN_POINT('', (66.0, -30.0, 90.099999999999994));
#5339=VERTEX_POINT('', #5338);
#5340=DIRECTION('', (0.0, 0.0, -1.0));
#5341=VECTOR('', #5340, 0.1999999999999989);
#5342=LINE('', #5338, #5341);
#5343=EDGE_CURVE('', #5339, #5330, #5342, .T.);
#5344=ORIENTED_EDGE('', *, *, #5343, .F.);
#5345=CARTESIAN_POINT('', (50.099999999999994, -30.0, 106.0));
#5346=VERTEX_POINT('', #5345);
#5347=CARTESIAN_POINT('', (50.099999999999994, -30.0, 90.099999999999994));
#5348=DIRECTION('', (0.0, 1.0, 0.0));
#5349=DIRECTION('', (0.0, 0.0, 1.0));
#5350=AXIS2_PLACEMENT_3D('', #5347, #5348, #5349);
#5351=CIRCLE('', #5350, 15.900000000000000);
#5352=EDGE_CURVE('', #5346, #5339, #5351, .T.);
#5353=ORIENTED_EDGE('', *, *, #5352, .F.);
#5354=DIRECTION('', (1.0, 0.0, 0.0));
#5355=VECTOR('', #5354, 50.099999999999994);
#5356=LINE('', #147, #5355);
#5357=EDGE_CURVE('', #148, #5346, #5356, .T.);
#5358=ORIENTED_EDGE('', *, *, #5357, .F.);
#5359=EDGE_LOOP('', (#5321, #5328, #5337, #5344, #5353, #5358));
#5360=FACE_BOUND('', #5359, .F.);
#5361=CARTESIAN_POINT('', (260.0, -30.0, 0.0));
#5362=DIRECTION('', (0.0, 1.0, 0.0));
#5363=DIRECTION('', (0.0, 0.0, 1.0));
#5364=AXIS2_PLACEMENT_3D('', #5361, #5362, #5363);
#5365=PLANE('', #5364);
#5366=ADVANCED_FACE('', (#5360), #5365, .T.);
#5367=SHAPE_ASPECT('featured shape: face 161', $, #38, .T.);
#5368=PROPERTY_DEFINITION('', $, #5367);
#5369=FACE_SHAPE_REPRESENTATION('', (#5366), #29);

```

```

#5370=PROPERTY_DEFINITION_REPRESENTATION(#5368,#5369);
#5371=ORIENTED_EDGE('',*,*,#154,.F.);
#5372=ORIENTED_EDGE('',*,*,#5357,.T.);
#5373=DIRECTION('',(0.0,-1.0,0.0));
#5374=VECTOR('',#5373,20.0);
#5375=LINE('',#5272,#5374);
#5376=EDGE_CURVE('',#5273,#5346,#5375,.T.);
#5377=ORIENTED_EDGE('',*,*,#5376,.F.);
#5378=ORIENTED_EDGE('',*,*,#5277,.F.);
#5379=EDGE_LOOP('',(#5371,#5372,#5377,#5378));
#5380=FACE_BOUND('',#5379,.F.);
#5381=CARTESIAN_POINT('',(0.0,-10.0,106.0));
#5382=DIRECTION('',(0.0,0.0,1.0));
#5383=DIRECTION('',(1.0,0.0,0.0));
#5384=AXIS2_PLACEMENT_3D('',#5381,#5382,#5383);
#5385=PLANE('',#5384);
#5386=ADVANCED_FACE('',(#5380),#5385,.F.);
#5387=SHAPE_ASPECT('featured shape: face 162',$,#38,.T.);
#5388=PROPERTY_DEFINITION('',$,#5387);
#5389=FACE_SHAPE_REPRESENTATION('',(#5386),#29);
#5390=PROPERTY_DEFINITION_REPRESENTATION(#5388,#5389);
#5391=ORIENTED_EDGE('',*,*,#5286,.F.);
#5392=ORIENTED_EDGE('',*,*,#5376,.T.);
#5393=ORIENTED_EDGE('',*,*,#5352,.T.);
#5394=DIRECTION('',(0.0,-1.0,0.0));
#5395=VECTOR('',#5394,20.0);
#5396=LINE('',#5279,#5395);
#5397=EDGE_CURVE('',#5280,#5339,#5396,.T.);
#5398=ORIENTED_EDGE('',*,*,#5397,.F.);
#5399=EDGE_LOOP('',(#5391,#5392,#5393,#5398));
#5400=FACE_BOUND('',#5399,.F.);
#5401=CARTESIAN_POINT('',(50.099999999999994,-10.0,90.099999999999994));
#5402=DIRECTION('',(0.0,1.0,0.0));
#5403=DIRECTION('',(0.0,0.0,1.0));
#5404=AXIS2_PLACEMENT_3D('',#5401,#5402,#5403);
#5405=CYLINDRICAL_SURFACE('',#5404,15.900000000000000);
#5406=ADVANCED_FACE('',(#5400),#5405,.F.);
#5407=SHAPE_ASPECT('featured shape: face 163',$,#38,.T.);
#5408=PROPERTY_DEFINITION('',$,#5407);
#5409=FACE_SHAPE_REPRESENTATION('',(#5406),#29);
#5410=PROPERTY_DEFINITION_REPRESENTATION(#5408,#5409);
#5411=ORIENTED_EDGE('',*,*,#5293,.F.);
#5412=ORIENTED_EDGE('',*,*,#5397,.T.);
#5413=ORIENTED_EDGE('',*,*,#5343,.T.);
#5414=DIRECTION('',(0.0,-1.0,0.0));
#5415=VECTOR('',#5414,20.0);
#5416=LINE('',#5288,#5415);
#5417=EDGE_CURVE('',#5289,#5330,#5416,.T.);
#5418=ORIENTED_EDGE('',*,*,#5417,.F.);
#5419=EDGE_LOOP('',(#5411,#5412,#5413,#5418));
#5420=FACE_BOUND('',#5419,.F.);
#5421=CARTESIAN_POINT('',(66.0,-10.0,90.099999999999994));
#5422=DIRECTION('',(1.0,0.0,0.0));

```

```

#5423=DIRECTION('', (0.0,0.0,-1.0));
#5424=AXIS2_PLACEMENT_3D('', #5421, #5422, #5423);
#5425=PLANE('', #5424);
#5426=ADVANCED_FACE('', (#5420), #5425, .F.);
#5427=SHAPE_ASPECT('featured shape: face 164', $, #38, .T.);
#5428=PROPERTY_DEFINITION('', $, #5427);
#5429=FACE_SHAPE_REPRESENTATION('', (#5426), #29);
#5430=PROPERTY_DEFINITION_REPRESENTATION(#5428, #5429);
#5431=ORIENTED_EDGE('', *, *, #5302, .F.);
#5432=ORIENTED_EDGE('', *, *, #5417, .T.);
#5433=ORIENTED_EDGE('', *, *, #5336, .T.);
#5434=DIRECTION('', (0.0,-1.0,0.0));
#5435=VECTOR('', #5434, 20.0);
#5436=LINE('', #5295, #5435);
#5437=EDGE_CURVE('', #5296, #5323, #5436, .T.);
#5438=ORIENTED_EDGE('', *, *, #5437, .F.);
#5439=EDGE_LOOP('', (#5431, #5432, #5433, #5438));
#5440=FACE_BOUND('', #5439, .F.);
#5441=CARTESIAN_POINT('', (50.099999999999994, -10.0, 89.900000000000006));
#5442=DIRECTION('', (0.0,1.0,0.0));
#5443=DIRECTION('', (0.0,0.0,1.0));
#5444=AXIS2_PLACEMENT_3D('', #5441, #5442, #5443);
#5445=CYLINDRICAL_SURFACE('', #5444, 15.900000000000000);
#5446=ADVANCED_FACE('', (#5440), #5445, .F.);
#5447=SHAPE_ASPECT('featured shape: face 165', $, #38, .T.);
#5448=PROPERTY_DEFINITION('', $, #5447);
#5449=FACE_SHAPE_REPRESENTATION('', (#5446), #29);
#5450=PROPERTY_DEFINITION_REPRESENTATION(#5448, #5449);
#5451=ORIENTED_EDGE('', *, *, #224, .T.);
#5452=ORIENTED_EDGE('', *, *, #5307, .F.);
#5453=ORIENTED_EDGE('', *, *, #5437, .T.);
#5454=ORIENTED_EDGE('', *, *, #5327, .T.);
#5455=EDGE_LOOP('', (#5451, #5452, #5453, #5454));
#5456=FACE_BOUND('', #5455, .F.);
#5457=CARTESIAN_POINT('', (50.099999999999994, -10.0, 74.0));
#5458=DIRECTION('', (0.0,0.0,-1.0));
#5459=DIRECTION('', (-1.0,0.0,0.0));
#5460=AXIS2_PLACEMENT_3D('', #5457, #5458, #5459);
#5461=PLANE('', #5460);
#5462=ADVANCED_FACE('', (#5456), #5461, .F.);
#5463=SHAPE_ASPECT('featured shape: face 166', $, #38, .T.);
#5464=PROPERTY_DEFINITION('', $, #5463);
#5465=FACE_SHAPE_REPRESENTATION('', (#5462), #29);
#5466=PROPERTY_DEFINITION_REPRESENTATION(#5464, #5465);
#5467=CLOSED_SHELL('', (#74, #142, #238, #334, #425, #516, #607, #647, #1109, #1180, #12
60, #1360, #1796, #1816, #1836, #1856, #1876, #1900, #1916, #1940, #1956, #1990, #2019, #2
041, #2058, #2084, #2110, #2134, #2153, #2177, #2196, #2220, #2239, #2263, #2282, #2316, #
2364, #2388, #2404, #2434, #2468, #2508, #2532, #2548, #2564, #2598, #2638, #2662, #2678,
#2694, #2728, #2768, #2792, #2808, #2824, #2858, #2898, #2922, #2938, #2954, #2988, #3028
, #3052, #3068, #3084, #3118, #3158, #3182, #3198, #3214, #3240, #3268, #3288, #3304, #334
8, #3372, #3392, #3412, #3428, #3458, #3482, #3498, #3528, #3552, #3568, #3598, #3622, #36
38, #3668, #3692, #3708, #3724, #3740, #3756, #3772, #3788, #3804, #3820, #3836, #3852, #3
868, #3916, #3948, #3984, #4016, #4052, #4084, #4120, #4144, #4192, #4214, #4242, #4264, #

```

ISO 10303-223:2008 (E)

```

4286,#4308,#4323,#4338,#4360,#4388,#4410,#4432,#4454,#4469,#4484,#4506,#4622,
#4642,#4662,#4682,#4702,#4718,#4742,#4762,#4782,#4802,#4822,#4842,#4862,#4878
,#4902,#4922,#4942,#4966,#4986,#5006,#5030,#5050,#5070,#5094,#5114,#5134,#515
0,#5166,#5182,#5198,#5214,#5230,#5246,#5262,#5316,#5366,#5386,#5406,#5426,#54
46,#5462));
#5468=MANIFOLD_SOLID_BREP('',#5467);
#5469=ADVANCED_BREP_SHAPE_REPRESENTATION('',(#5468),#29);
#5470=SHAPE_DEFINITION_REPRESENTATION(#38,#5469);
#5471=DATUM_FEATURE('',$, #38,.T.);
#5472=PROPERTY_DEFINITION('',$, #5471);
#5473=REPRESENTATION('',(#5316),#29);
#5474=PROPERTY_DEFINITION_REPRESENTATION(#5472,#5473);
#5475=DATUM('DTM1',$, #38,.T.,'DTM1');
#5476=SHAPE_ASPECT_RELATIONSHIP('',$, #5471,#5475);
#5477=DATUM_REFERENCE(1,#5475);
#5478=DIMENSIONAL_LOCATION('',$, #1110,#5317);
#5479=PRECISION_QUALIFIER(4);
#5485=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5479))REPRESENTAT
ION_ITEM(''));
#5486=SHAPE_DIMENSION_REPRESENTATION('',(#5485),#29);
#5487=PRECISION_QUALIFIER(4);
#5488=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#5489=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#5490=MEASURE_QUALIFICATION('','',#5488,(#5487));
#5491=MEASURE_QUALIFICATION('','',#5489,(#5487));
#5492=TOLERANCE_VALUE(#5489,#5488);
#5493=PLUS_MINUS_TOLERANCE(#5492,#5478);
#5494=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5478,#5486);
#5495=CARTESIAN_POINT('',(0.0,0.0,90.0));
#5496=DIRECTION('',(0.0,0.0,-1.0));
#5497=DIRECTION('',(-1.0,0.0,0.0));
#5498=AXIS2_PLACEMENT_3D('DTM2',#5495,#5496,#5497);
#5499=DATUM_FEATURE('',$, #38,.T.);
#5500=PROPERTY_DEFINITION('',$, #5499);
#5501=REPRESENTATION('',(#5498),#29);
#5502=PROPERTY_DEFINITION_REPRESENTATION(#5500,#5501);
#5503=DATUM('DTM2',$, #38,.F.,'DTM2');
#5504=SHAPE_ASPECT_RELATIONSHIP('',$, #5499,#5503);
#5505=DATUM_REFERENCE(1,#5503);
#5506=DIMENSIONAL_LOCATION('',$, #75,#5508);
#5507=PRECISION_QUALIFIER(4);
#5508=SHAPE_ASPECT('',$, #38,.T.);
#5509=PROPERTY_DEFINITION('',$, #5508);
#5510=PROPERTY_DEFINITION_REPRESENTATION(#5509,#5501);
#5513=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5507))REPRESENTAT
ION_ITEM(''));
#5514=SHAPE_DIMENSION_REPRESENTATION('',(#5513),#29);
#5515=PRECISION_QUALIFIER(4);
#5516=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1500000000000000),#10);
#5517=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1500000000000000),#10);
#5518=MEASURE_QUALIFICATION('','',#5516,(#5515));

```



```

#5519=MEASURE_QUALIFICATION('', '#5517', (#5515));
#5520=TOLERANCE_VALUE(#5517, #5516);
#5521=PLUS_MINUS_TOLERANCE(#5520, #5506);
#5522=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5506, #5514);
#5523=DATUM_FEATURE('', $, #38, .T.);
#5524=PROPERTY_DEFINITION('', $, #5523);
#5525=REPRESENTATION('', (#334), #29);
#5526=PROPERTY_DEFINITION_REPRESENTATION(#5524, #5525);
#5527=DATUM('A', $, #38, .T., 'A');
#5528=SHAPE_ASPECT_RELATIONSHIP('', $, #5523, #5527);
#5529=DATUM_REFERENCE(1, #5527);
#5530=DATUM_FEATURE('', $, #38, .T.);
#5531=PROPERTY_DEFINITION('', $, #5530);
#5532=REPRESENTATION('', (#238), #29);
#5533=PROPERTY_DEFINITION_REPRESENTATION(#5531, #5532);
#5534=DATUM('C', $, #38, .T., 'C');
#5535=SHAPE_ASPECT_RELATIONSHIP('', $, #5530, #5534);
#5536=DATUM_REFERENCE(1, #5534);
#5537=DATUM_FEATURE('', $, #38, .T.);
#5538=PROPERTY_DEFINITION('', $, #5537);
#5539=REPRESENTATION('', (#1109), #29);
#5540=PROPERTY_DEFINITION_REPRESENTATION(#5538, #5539);
#5541=DATUM('D', $, #38, .T., 'D');
#5542=SHAPE_ASPECT_RELATIONSHIP('', $, #5537, #5541);
#5543=DATUM_REFERENCE(1, #5541);
#5544=DATUM_FEATURE('', $, #38, .T.);
#5545=PROPERTY_DEFINITION('', $, #5544);
#5546=REPRESENTATION('', (#2110), #29);
#5547=PROPERTY_DEFINITION_REPRESENTATION(#5545, #5546);
#5548=DATUM('F', $, #38, .T., 'F');
#5549=SHAPE_ASPECT_RELATIONSHIP('', $, #5544, #5548);
#5550=DATUM_REFERENCE(1, #5548);
#5551=DATUM_FEATURE('', $, #38, .T.);
#5552=PROPERTY_DEFINITION('', $, #5551);
#5553=REPRESENTATION('', (#1260), #29);
#5554=PROPERTY_DEFINITION_REPRESENTATION(#5552, #5553);
#5555=DATUM('E', $, #38, .T., 'E');
#5556=SHAPE_ASPECT_RELATIONSHIP('', $, #5551, #5555);
#5557=DATUM_REFERENCE(1, #5555);

```

```

#5595=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() SHAPE
_ASPECT('', $, #38, .T.) STEP());
#5596=PROPERTY_DEFINITION('', $, #5595);
#5597=PRODUCT_DEFINITION_SHAPE('', $, #5595);
#5598=CARTESIAN_POINT('', (230.0, -20.0, 0.0));
#5599=DIRECTION('', (0.0, 0.0, 1.0));
#5600=DIRECTION('', (0.707106781186547, 0.707106781186547, 0.0));
#5601=AXIS2_PLACEMENT_3D('orientation', #5598, #5599, #5600);
#5602=REPRESENTATION('', (#1260, #1360), #29);
#5603=PROPERTY_DEFINITION_REPRESENTATION(#5596, #5602);
#5604=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5601), #29);
#5605=PROPERTY_DEFINITION_REPRESENTATION(#5596, #5604);

```

```

#5606=FEATURE_COMPONENT_DEFINITION('', $);
#5607=PRODUCT_DEFINITION_SHAPE('', $, #5606);
#5608=PATH_FEATURE_COMPONENT('linear path', 'linear', #5607, .F.);
#5609=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.3000000000000000);
#5610=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.3000000000000000);
#5611=PRECISION_QUALIFIER(4);
#5617=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(180.0), #10)

QUALIFIED_REPRESENTATION_ITEM((#5609, #5610, #5611))REPRESENTATION_ITEM('distan
ce'));
#5618=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5601, #5617), #29);
#5619=PROPERTY_DEFINITION('', $, #5608);
#5620=PROPERTY_DEFINITION_REPRESENTATION(#5619, #5618);
#5621=DIRECTION_SHAPE_REPRESENTATION('', (#5599), #29);
#5622=PROPERTY_DEFINITION_REPRESENTATION(#5619, #5621);
#5623=FEATURE_COMPONENT_DEFINITION('', $);
#5624=PRODUCT_DEFINITION_SHAPE('', $, #5623);
#5625=VEE_PROFILE('vee profile', 'v-shape', #5624, .F.);
#5630=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19)
    PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5635=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0), #19)
    PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5636=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5601, #5630, #5635), #29);
#5637=PROPERTY_DEFINITION('', $, #5625);
#5638=PROPERTY_DEFINITION_REPRESENTATION(#5637, #5636);
#5639=SHAPE_ASPECT('', 'removal boundary occurrence', #5597, .T.);
#5640=SHAPE_DEFINING_RELATIONSHIP('vee boundary', 'profile
usage', #5625, #5639);
#5641=SHAPE_ASPECT('', 'course of travel occurrence', #5597, .T.);
#5642=SHAPE_DEFINING_RELATIONSHIP('course of travel', 'path feature component
usage', #5608, #5641);
#5643=DIMENSIONAL_LOCATION('', $, #1110, #1361);
#5644=PRECISION_QUALIFIER(4);
#5650=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5644))REPRESENTAT
ION_ITEM(''));
#5651=SHAPE_DIMENSION_REPRESENTATION('', (#5650), #29);
#5652=PRECISION_QUALIFIER(4);
#5653=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#5654=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000), #10);
#5655=MEASURE_QUALIFICATION('', '#5653, (#5652)');
#5656=MEASURE_QUALIFICATION('', '#5654, (#5652)');
#5657=TOLERANCE_VALUE(#5654, #5653);
#5658=PLUS_MINUS_TOLERANCE(#5657, #5643);
#5659=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5643, #5651);
#5660=DIMENSIONAL_LOCATION('', $, #239, #1261);
#5661=PRECISION_QUALIFIER(4);
#5667=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5661))REPRESENTA
TION_ITEM(''));

```

```

#5668=SHAPE_DIMENSION_REPRESENTATION('', (#5667), #29);
#5669=PRECISION_QUALIFIER(4);
#5670=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#5671=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#5672=MEASURE_QUALIFICATION('', '#5670', (#5669));
#5673=MEASURE_QUALIFICATION('', '#5671', (#5669));
#5674=TOLERANCE_VALUE(#5671, #5670);
#5675=PLUS_MINUS_TOLERANCE(#5674, #5660);
#5676=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5660, #5668);

#5682=(CHARACTERIZED_OBJECT('V_LIN', $) FEATURE_DEFINITION() INSTANCED_FEATURE()
REMOVAL_VOLUME() SHAPE_ASPECT('V_LIN', $, #38, .T.));
#5683=PROPERTY_DEFINITION('', $, #5682);
#5684=PRODUCT_DEFINITION_SHAPE('', $, #5682);
#5685=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#5686=DIRECTION('', (0.0, 0.0, 1.0));
#5687=DIRECTION('', (1.0, 0.0, 0.0));
#5688=AXIS2_PLACEMENT_3D('orientation', #5685, #5686, #5687);
#5689=REPRESENTATION('', (#1260, #1796), #29);
#5690=PROPERTY_DEFINITION_REPRESENTATION(#5683, #5689);
#5691=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5688), #29);
#5692=PROPERTY_DEFINITION_REPRESENTATION(#5683, #5691);
#5693=SHAPE_ASPECT('', 'volume shape', #38, .T.);
#5694=PROPERTY_DEFINITION('', $, #5693);
#5695=PROPERTY_DEFINITION_REPRESENTATION(#5694, #5689);
#5696=SHAPE_ASPECT('', 'shape volume occurrence', #5684, .T.);
#5697=SHAPE_DEFINING_RELATIONSHIP('', 'volume shape usage', #5693, #5696);
#5703=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() SHAPE
_ASPECT('', $, #38, .T.) STEP());
#5704=PROPERTY_DEFINITION('', $, #5703);
#5705=PRODUCT_DEFINITION_SHAPE('', $, #5703);
#5706=CARTESIAN_POINT('', (230.0, -10.0, 130.0));
#5707=DIRECTION('', (0.0, -1.0, 0.0));
#5708=DIRECTION('', (0.707106781186547, 0.0, 0.707106781186547));
#5709=AXIS2_PLACEMENT_3D('orientation', #5706, #5707, #5708);
#5710=REPRESENTATION('', (#1260, #1816, #1836), #29);
#5711=PROPERTY_DEFINITION_REPRESENTATION(#5704, #5710);
#5712=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5709), #29);
#5713=PROPERTY_DEFINITION_REPRESENTATION(#5704, #5712);
#5714=FEATURE_COMPONENT_DEFINITION('', $);
#5715=PRODUCT_DEFINITION_SHAPE('', $, #5714);
#5716=PATH_FEATURE_COMPONENT('linear path', 'linear', #5715, .F.);
#5717=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#5718=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#5719=PRECISION_QUALIFIER(4);
#5725=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0), #10) QUALIFIED_REPRESENTATION_ITEM((#5717, #5718, #5719)
) REPRESENTATION_ITEM('distance'));
#5726=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5709, #5725), #29);
#5727=PROPERTY_DEFINITION('', $, #5716);
#5728=PROPERTY_DEFINITION_REPRESENTATION(#5727, #5726);
#5729=DIRECTION_SHAPE_REPRESENTATION('', (#5707), #29);
#5730=PROPERTY_DEFINITION_REPRESENTATION(#5727, #5729);

```

```

#5731=FEATURE_COMPONENT_DEFINITION('', $);
#5732=PRODUCT_DEFINITION_SHAPE('', $, #5731);
#5733=VEE_PROFILE('vee profile', 'v-shape', #5732, .F.);
#5738=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5743=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-45.0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5744=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#5745=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#5746=PRECISION_QUALIFIER(4);
#5752=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5744, #5745, #5746)REPRESENTATION_ITEM('profile radius'));
#5753=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5709, #5738, #5743, #5752), #29);
#5754=PROPERTY_DEFINITION('', $, #5733);
#5755=PROPERTY_DEFINITION_REPRESENTATION(#5754, #5753);
#5756=SHAPE_ASPECT('', 'removal boundary occurrence', #5705, .T.);
#5757=SHAPE_DEFINING_RELATIONSHIP('vee boundary', 'profile usage', #5733, #5756);
#5758=SHAPE_ASPECT('', 'course of travel occurrence', #5705, .T.);
#5759=SHAPE_DEFINING_RELATIONSHIP('course of travel', 'path feature component usage', #5716, #5758);
#5760=DIMENSIONAL_LOCATION('', $, #143, #1837);
#5761=PRECISION_QUALIFIER(4);
#5767=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(50.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5761)REPRESENTATION_ITEM(''));
#5768=SHAPE_DIMENSION_REPRESENTATION('', (#5767), #29);
#5769=PRECISION_QUALIFIER(4);
#5770=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000), #10);
#5771=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.3000000000000000), #10);
#5772=MEASURE_QUALIFICATION('', '#5770, (#5769)');
#5773=MEASURE_QUALIFICATION('', '#5771, (#5769)');
#5774=TOLERANCE_VALUE(#5771, #5770);
#5775=PLUS_MINUS_TOLERANCE(#5774, #5760);
#5776=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5760, #5768);
#5777=CARTESIAN_POINT('', (230.0, -10.0, 180.0));
#5778=DIRECTION('', (-1.0, 0.0, 0.0));
#5779=DIRECTION('', (0.0, 0.0, -1.0));
#5780=AXIS2_PLACEMENT_3D('', #5777, #5778, #5779);
#5781=SHAPE_ASPECT('', $, #38, .F.);
#5782=PROPERTY_DEFINITION('', $, #5781);
#5783=REPRESENTATION('', (#5780), #29);
#5784=PROPERTY_DEFINITION_REPRESENTATION(#5782, #5783);
#5785=DIMENSIONAL_LOCATION('', $, #239, #5781);
#5786=PRECISION_QUALIFIER(4);
#5792=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(230.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5786)REPRESENTATION_ITEM(''));
#5793=SHAPE_DIMENSION_REPRESENTATION('', (#5792), #29);
#5794=PRECISION_QUALIFIER(4);
#5795=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#5796=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);

```

```

#5797=MEASURE_QUALIFICATION('', '#5795, (#5794));
#5798=MEASURE_QUALIFICATION('', '#5796, (#5794));
#5799=TOLERANCE_VALUE(#5796, #5795);
#5800=PLUS_MINUS_TOLERANCE(#5799, #5785);
#5801=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5785, #5793);
#5807=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('', $, #38, .T.)STEP());
#5808=PROPERTY_DEFINITION('', $, #5807);
#5809=PRODUCT_DEFINITION_SHAPE('', $, #5807);
#5810=CARTESIAN_POINT('', (230.0, -50.0, 50.0));
#5811=DIRECTION('', (0.0, 1.0, 0.0));
#5812=DIRECTION('', (0.707106781186547, 0.0, -0.707106781186547));
#5813=AXIS2_PLACEMENT_3D('orientation', #5810, #5811, #5812);
#5814=REPRESENTATION('', (#1260, #1856, #1876), #29);
#5815=PROPERTY_DEFINITION_REPRESENTATION(#5808, #5814);
#5816=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5813), #29);
#5817=PROPERTY_DEFINITION_REPRESENTATION(#5808, #5816);
#5818=FEATURE_COMPONENT_DEFINITION('', $);
#5819=PRODUCT_DEFINITION_SHAPE('', $, #5818);
#5820=PATH_FEATURE_COMPONENT('linear path', 'linear', #5819, .F.);
#5821=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.100000000000000);
#5822=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.100000000000000);
#5823=PRECISION_QUALIFIER(4);
#5829=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5821, #5822, #5823)
)REPRESENTATION_ITEM('distance'));
#5830=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5813, #5829), #29);
#5831=PROPERTY_DEFINITION('', $, #5820);
#5832=PROPERTY_DEFINITION_REPRESENTATION(#5831, #5830);
#5833=DIRECTION_SHAPE_REPRESENTATION('', (#5811), #29);
#5834=PROPERTY_DEFINITION_REPRESENTATION(#5831, #5833);
#5835=FEATURE_COMPONENT_DEFINITION('', $);
#5836=PRODUCT_DEFINITION_SHAPE('', $, #5835);
#5837=VEE_PROFILE('vee profile', 'v-shape', #5836, .F.);
#5842=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#5847=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#5848=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.100000000000000);
#5849=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.100000000000000);
#5850=PRECISION_QUALIFIER(4);
#5856=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5848, #5849, #5850)
)REPRESENTATION_ITEM('profile radius'));
#5857=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5813, #5842, #5847, #5856), #29);
#5858=PROPERTY_DEFINITION('', $, #5837);
#5859=PROPERTY_DEFINITION_REPRESENTATION(#5858, #5857);
#5860=SHAPE_ASPECT('', 'removal boundary occurrence', #5809, .T.);
#5861=SHAPE_DEFINING_RELATIONSHIP('vee boundary', 'profile
usage', #5837, #5860);
#5862=SHAPE_ASPECT('', 'course of travel occurrence', #5809, .T.);
#5863=SHAPE_DEFINING_RELATIONSHIP('course of travel', 'path feature component
usage', #5820, #5862);

```

```

#5864=DIMENSIONAL_LOCATION('', $, #75, #1877);
#5865=PRECISION_QUALIFIER(4);
#5871=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5865))REPRESENTAT
ION_ITEM(''));
#5872=SHAPE_DIMENSION_REPRESENTATION('', (#5871), #29);
#5873=PRECISION_QUALIFIER(4);
#5874=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000), #10);
#5875=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.3000000000000000), #10);
#5876=MEASURE_QUALIFICATION('', '#5874, (#5873));
#5877=MEASURE_QUALIFICATION('', '#5875, (#5873));
#5878=TOLERANCE_VALUE(#5875, #5874);
#5879=PLUS_MINUS_TOLERANCE(#5878, #5864);
#5880=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5864, #5872);
#5881=CARTESIAN_POINT('', (230.0, -50.0, 0.0));
#5882=DIRECTION('', (-1.0, 0.0, 0.0));
#5883=DIRECTION('', (0.0, 0.0, 1.0));
#5884=AXIS2_PLACEMENT_3D('', #5881, #5882, #5883);
#5885=SHAPE_ASPECT('', $, #38, .F.);
#5886=PROPERTY_DEFINITION('', $, #5885);
#5887=REPRESENTATION('', (#5884), #29);
#5888=PROPERTY_DEFINITION_REPRESENTATION(#5886, #5887);
#5889=DIMENSIONAL_LOCATION('', $, #239, #5885);
#5890=PRECISION_QUALIFIER(4);
#5896=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(230.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5890))REPRESENTA
TION_ITEM(''));
#5897=SHAPE_DIMENSION_REPRESENTATION('', (#5896), #29);
#5898=PRECISION_QUALIFIER(4);
#5899=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#5900=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#5901=MEASURE_QUALIFICATION('', '#5899, (#5898));
#5902=MEASURE_QUALIFICATION('', '#5900, (#5898));
#5903=TOLERANCE_VALUE(#5900, #5899);
#5904=PLUS_MINUS_TOLERANCE(#5903, #5889);
#5905=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5889, #5897);
#5911=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('', $, #38, .T.);
#5912=PROPERTY_DEFINITION('', $, #5911);
#5913=PRODUCT_DEFINITION_SHAPE('', $, #5911);
#5914=CARTESIAN_POINT('', (245.0, -20.0, 110.0));
#5915=DIRECTION('', (0.0, -1.0, 0.0));
#5916=DIRECTION('', (1.0, 0.0, 0.0));
#5917=AXIS2_PLACEMENT_3D('orientation', #5914, #5915, #5916);
#5918=REPRESENTATION('', (#1900, #1916), #29);
#5919=PROPERTY_DEFINITION_REPRESENTATION(#5912, #5918);
#5920=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#5917), #29);
#5921=PROPERTY_DEFINITION_REPRESENTATION(#5912, #5920);
#5922=FEATURE_COMPONENT_DEFINITION('', $);
#5923=PRODUCT_DEFINITION_SHAPE('', $, #5922);
#5924=CIRCULAR_CLOSED_PROFILE('circular profile', $, #5923, .F.);
#5925=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#5926=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);

```

```

#5927=PRECISION_QUALIFIER(4);
#5933=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5925,#5926,#5927)
)REPRESENTATION_ITEM('diameter'));
#5934=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5917,#5933),#29);
#5935=PROPERTY_DEFINITION('',$,#5924);
#5936=PROPERTY_DEFINITION_REPRESENTATION(#5935,#5934);
#5937=FEATURE_COMPONENT_DEFINITION('',$);
#5938=PRODUCT_DEFINITION_SHAPE('',$,#5937);
#5939=PATH_FEATURE_COMPONENT('linear path','linear',#5938,.F.);
#5944=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)REPRESENTATION_ITEM('distance'));
#5945=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#5917,#5944),#29);
#5946=PROPERTY_DEFINITION('',$,#5939);
#5947=PROPERTY_DEFINITION_REPRESENTATION(#5946,#5945);
#5948=DIRECTION_SHAPE_REPRESENTATION('',(#5915),#29);
#5949=PROPERTY_DEFINITION_REPRESENTATION(#5946,#5948);
#5950=SHAPE_ASPECT('','diameter occurrence',#5913,.T.);
#5951=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#5924,#5950);
#5952=SHAPE_ASPECT('','hole depth occurrence',#5913,.T.);
#5953=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#5939,#5952);
#5954=FEATURE_COMPONENT_DEFINITION('',$);
#5955=PRODUCT_DEFINITION_SHAPE('',$,#5954);
#5956=HOLE_BOTTOM('bottom condition','through',#5955,.F.);
#5957=SHAPE_ASPECT('','bottom condition occurrence',#5913,.T.);
#5958=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#5956,#5957);
#5959=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#5960=DIRECTION('',(0.0,-1.0,0.0));
#5961=AXIS1_PLACEMENT('',#5959,#5960);
#5962=SHAPE_ASPECT('',$,#38,.F.);
#5963=PROPERTY_DEFINITION('',$,#5962);
#5964=REPRESENTATION('',(#5961),#29);
#5965=PROPERTY_DEFINITION_REPRESENTATION(#5963,#5964);
#5966=DIMENSIONAL_LOCATION('',$,#648,#5962);
#5967=PRECISION_QUALIFIER(4);
#5973=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0),#10)QUALIFIED_REPRESENTATION_ITEM((#5967)REPRESENTAT
ION_ITEM(''));
#5974=SHAPE_DIMENSION_REPRESENTATION('',(#5973),#29);
#5975=PRECISION_QUALIFIER(4);
#5976=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#5977=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#5978=MEASURE_QUALIFICATION('','',#5976,(#5975));
#5979=MEASURE_QUALIFICATION('','',#5977,(#5975));
#5980=TOLERANCE_VALUE(#5977,#5976);
#5981=PLUS_MINUS_TOLERANCE(#5980,#5966);
#5982=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5966,#5974);
#5983=CARTESIAN_POINT('',(245.0,-20.0,110.0));
#5984=DIRECTION('',(0.0,-1.0,0.0));
#5985=AXIS1_PLACEMENT('',#5983,#5984);
#5986=SHAPE_ASPECT('',$,#38,.F.);

```

```

#5987=PROPERTY_DEFINITION('', $, #5986);
#5988=REPRESENTATION('', (#5985), #29);
#5989=PROPERTY_DEFINITION_REPRESENTATION(#5987, #5988);
#5990=DIMENSIONAL_LOCATION('', $, #1837, #5986);
#5991=PRECISION_QUALIFIER(4);
#5997=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10)QUALIFIED_REPRESENTATION_ITEM((#5991))REPRESENTAT
ION_ITEM(''));
#5998=SHAPE_DIMENSION_REPRESENTATION('', (#5997), #29);
#5999=PRECISION_QUALIFIER(4);
#6000=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#6001=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#6002=MEASURE_QUALIFICATION('', '#6000, (#5999);
#6003=MEASURE_QUALIFICATION('', '#6001, (#5999);
#6004=TOLERANCE_VALUE(#6001, #6000);
#6005=PLUS_MINUS_TOLERANCE(#6004, #5990);
#6006=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#5990, #5998);
#6012=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('', $, #38, .T.);
#6013=PROPERTY_DEFINITION('', $, #6012);
#6014=PRODUCT_DEFINITION_SHAPE('', $, #6012);
#6015=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#6016=DIRECTION('', (0.0, -1.0, 0.0));
#6017=DIRECTION('', (1.0, 0.0, 0.0));
#6018=AXIS2_PLACEMENT_3D('orientation', #6015, #6016, #6017);
#6019=REPRESENTATION('', (#1940, #1956), #29);
#6020=PROPERTY_DEFINITION_REPRESENTATION(#6013, #6019);
#6021=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6018), #29);
#6022=PROPERTY_DEFINITION_REPRESENTATION(#6013, #6021);
#6023=FEATURE_COMPONENT_DEFINITION('', $);
#6024=PRODUCT_DEFINITION_SHAPE('', $, #6023);
#6025=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6024, .F.);
#6026=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#6027=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#6028=PRECISION_QUALIFIER(4);
#6034=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0), #10)QUALIFIED_REPRESENTATION_ITEM((#6026, #6027, #6028)
)REPRESENTATION_ITEM('diameter'));
#6035=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6018, #6034), #29);
#6036=PROPERTY_DEFINITION('', $, #6025);
#6037=PROPERTY_DEFINITION_REPRESENTATION(#6036, #6035);
#6038=FEATURE_COMPONENT_DEFINITION('', $);
#6039=PRODUCT_DEFINITION_SHAPE('', $, #6038);
#6040=PATH_FEATURE_COMPONENT('linear path', 'linear', #6039, .F.);
#6045=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10)REPRESENTATION_ITEM('distance'));
#6046=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6018, #6045), #29);
#6047=PROPERTY_DEFINITION('', $, #6040);
#6048=PROPERTY_DEFINITION_REPRESENTATION(#6047, #6046);
#6049=DIRECTION_SHAPE_REPRESENTATION('', (#6016), #29);
#6050=PROPERTY_DEFINITION_REPRESENTATION(#6047, #6049);
#6051=SHAPE_ASPECT('', 'diameter occurrence', #6014, .T.);
#6052=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6025, #6051);

```



```

#6053=SHAPE_ASPECT('', 'hole depth occurrence', #6014, .T.);
#6054=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6040, #6053);
#6055=FEATURE_COMPONENT_DEFINITION('', $);
#6056=PRODUCT_DEFINITION_SHAPE('', $, #6055);
#6057=HOLE_BOTTOM('bottom condition', 'through', #6056, .F.);
#6058=SHAPE_ASPECT('', 'bottom condition occurrence', #6014, .T.);
#6059=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #6057, #6058);
#6060=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#6061=DIRECTION('', (0.0, -1.0, 0.0));
#6062=AXIS1_PLACEMENT('', #6060, #6061);
#6063=SHAPE_ASPECT('', $, #38, .F.);
#6064=PROPERTY_DEFINITION('', $, #6063);
#6065=REPRESENTATION('', (#6062), #29);
#6066=PROPERTY_DEFINITION_REPRESENTATION(#6064, #6065);
#6067=DIMENSIONAL_LOCATION('', $, #648, #6063);
#6068=PRECISION_QUALIFIER(4);
#6074=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6068)) REPRESENTAT
ION_ITEM(''));
#6075=SHAPE_DIMENSION_REPRESENTATION('', (#6074), #29);
#6076=PRECISION_QUALIFIER(4);
#6077=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#6078=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#6079=MEASURE_QUALIFICATION('', '#6077, (#6076)');
#6080=MEASURE_QUALIFICATION('', '#6078, (#6076)');
#6081=TOLERANCE_VALUE(#6078, #6077);
#6082=PLUS_MINUS_TOLERANCE(#6081, #6067);
#6083=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6067, #6075);
#6084=CARTESIAN_POINT('', (245.0, -20.0, 70.0));
#6085=DIRECTION('', (0.0, -1.0, 0.0));
#6086=AXIS1_PLACEMENT('', #6084, #6085);
#6087=SHAPE_ASPECT('', $, #38, .F.);
#6088=PROPERTY_DEFINITION('', $, #6087);
#6089=REPRESENTATION('', (#6086), #29);
#6090=PROPERTY_DEFINITION_REPRESENTATION(#6088, #6089);
#6091=DIMENSIONAL_LOCATION('', $, #1877, #6087);
#6092=PRECISION_QUALIFIER(4);
#6098=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6092)) REPRESENTAT
ION_ITEM(''));
#6099=SHAPE_DIMENSION_REPRESENTATION('', (#6098), #29);
#6100=PRECISION_QUALIFIER(4);
#6101=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#6102=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#6103=MEASURE_QUALIFICATION('', '#6101, (#6100)');
#6104=MEASURE_QUALIFICATION('', '#6102, (#6100)');
#6105=TOLERANCE_VALUE(#6102, #6101);
#6106=PLUS_MINUS_TOLERANCE(#6105, #6091);
#6107=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6091, #6099);
#6113=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));

```

```

#6114=PROPERTY_DEFINITION('', $, #6113);
#6115=PRODUCT_DEFINITION_SHAPE('', $, #6113);
#6116=CARTESIAN_POINT('', (160.0, 0.0, 90.0));
#6117=DIRECTION('', (0.0, -1.0, 0.0));
#6118=DIRECTION('', (1.0, 0.0, 0.0));
#6119=AXIS2_PLACEMENT_3D('orientation', #6116, #6117, #6118);
#6120=REPRESENTATION('', (#1990, #2019, #2041, #2058), #29);
#6121=PROPERTY_DEFINITION_REPRESENTATION(#6114, #6120);
#6122=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6119), #29);
#6123=PROPERTY_DEFINITION_REPRESENTATION(#6114, #6122);
#6124=FEATURE_COMPONENT_DEFINITION('', $);
#6125=PRODUCT_DEFINITION_SHAPE('', $, #6124);
#6126=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6125, .F.);
#6127=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#6128=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#6129=PRECISION_QUALIFIER(4);
#6135=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0), #10)QUALIFIED_REPRESENTATION_ITEM((#6127, #6128, #6129)
)REPRESENTATION_ITEM('diameter'));
#6136=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6119, #6135), #29);
#6137=PROPERTY_DEFINITION('', $, #6126);
#6138=PROPERTY_DEFINITION_REPRESENTATION(#6137, #6136);
#6139=FEATURE_COMPONENT_DEFINITION('', $);
#6140=PRODUCT_DEFINITION_SHAPE('', $, #6139);
#6141=PATH_FEATURE_COMPONENT('linear path', 'linear', #6140, .F.);
#6146=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0000000000000007), #10)REPRESENTATION_ITEM('distance'));
#6147=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6119, #6146), #29);
#6148=PROPERTY_DEFINITION('', $, #6141);
#6149=PROPERTY_DEFINITION_REPRESENTATION(#6148, #6147);
#6150=DIRECTION_SHAPE_REPRESENTATION('', (#6117), #29);
#6151=PROPERTY_DEFINITION_REPRESENTATION(#6148, #6150);
#6152=SHAPE_ASPECT('', 'diameter occurrence', #6115, .T.);
#6153=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6126, #6152);
#6154=SHAPE_ASPECT('', 'hole depth occurrence', #6115, .T.);
#6155=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6141, #6154);
#6156=FEATURE_COMPONENT_DEFINITION('', $);
#6157=PRODUCT_DEFINITION_SHAPE('', $, #6156);
#6158=HOLE_BOTTOM('bottom condition', 'through', #6157, .F.);
#6159=SHAPE_ASPECT('', 'bottom condition occurrence', #6115, .T.);
#6160=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #6158, #6159);
#6161=CARTESIAN_POINT('', (160.0, 0.0, 90.0));
#6162=DIRECTION('', (0.0, -1.0, 0.0));
#6163=AXIS1_PLACEMENT('', #6161, #6162);
#6164=SHAPE_ASPECT('', $, #38, .F.);
#6165=PROPERTY_DEFINITION('', $, #6164);
#6166=REPRESENTATION('', (#6163), #29);
#6167=PROPERTY_DEFINITION_REPRESENTATION(#6165, #6166);
#6168=DIMENSIONAL_LOCATION('', $, #1261, #6164);
#6169=PRECISION_QUALIFIER(4);

```

```

#6175=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6169))REPRESENTAT
ION_ITEM(''));
#6176=SHAPE_DIMENSION_REPRESENTATION('',( #6175),#29);
#6177=PRECISION_QUALIFIER(4);
#6178=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#6179=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000),#10);
#6180=MEASURE_QUALIFICATION('','',#6178,(#6177));
#6181=MEASURE_QUALIFICATION('','',#6179,(#6177));
#6182=TOLERANCE_VALUE(#6179,#6178);
#6183=PLUS_MINUS_TOLERANCE(#6182,#6168);
#6184=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6168,#6176);
#6185=CARTESIAN_POINT('',(160.0,0.0,90.0));
#6186=DIRECTION('',(0.0,-1.0,0.0));
#6187=AXIS1_PLACEMENT('','#6185,#6186);
#6188=SHAPE_ASPECT('','$,#38,.F.);
#6189=PROPERTY_DEFINITION('','$,#6188);
#6190=REPRESENTATION('',( #6187),#29);
#6191=PROPERTY_DEFINITION_REPRESENTATION(#6189,#6190);
#6192=DIMENSIONAL_LOCATION('','$,#143,#6188);
#6193=PRECISION_QUALIFIER(4);
#6199=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6193))REPRESENTAT
ION_ITEM(''));
#6200=SHAPE_DIMENSION_REPRESENTATION('',( #6199),#29);
#6201=PRECISION_QUALIFIER(4);
#6202=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1500000000000000),#10);
#6203=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1500000000000000),#10);
#6204=MEASURE_QUALIFICATION('','',#6202,(#6201));
#6205=MEASURE_QUALIFICATION('','',#6203,(#6201));
#6206=TOLERANCE_VALUE(#6203,#6202);
#6207=PLUS_MINUS_TOLERANCE(#6206,#6192);
#6208=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6192,#6200);
#6214=(CHARACTERIZED_OBJECT('','open
rectangular')FEATURE_DEFINITION()INSTANCED_FEATURE()POCKET()SHAPE_ASPECT('','
open rectangular',#38,.T.));
#6215=PROPERTY_DEFINITION('','$,#6214);
#6216=PRODUCT_DEFINITION_SHAPE('','$,#6214);
#6217=CARTESIAN_POINT('',(50.0,0.0,90.0));
#6218=DIRECTION('',(0.0,-1.0,0.0));
#6219=DIRECTION('',(0.0,0.0,1.0));
#6220=AXIS2_PLACEMENT_3D('orientation',#6217,#6218,#6219);
#6221=REPRESENTATION('',( #2084,#2110),#29);
#6222=PROPERTY_DEFINITION_REPRESENTATION(#6215,#6221);
#6223=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #6220),#29);
#6224=PROPERTY_DEFINITION_REPRESENTATION(#6215,#6223);
#6225=FEATURE_COMPONENT_DEFINITION('','$);
#6226=PRODUCT_DEFINITION_SHAPE('','$,#6225);
#6227=SQUARE_U_PROFILE('square u profile','boundary',#6226,.F.);
#6228=STANDARD_UNCERTAINTY('upper limit','plus',0.1000000000000000);
#6229=STANDARD_UNCERTAINTY('lower limit','minus',-0.1000000000000000);
#6230=PRECISION_QUALIFIER(4);

```

```

#6236=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6228,#6229,#6230)
)REPRESENTATION_ITEM('width'));
#6241=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#6246=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#6247=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6220,#6236,#6241,#6246),#29);
#6248=PROPERTY_DEFINITION('',$,#6227);
#6249=PROPERTY_DEFINITION_REPRESENTATION(#6248,#6247);
#6250=FEATURE_COMPONENT_DEFINITION('',$);
#6251=PRODUCT_DEFINITION_SHAPE('',$,#6250);
#6252=PATH_FEATURE_COMPONENT('linear path','linear',#6251,.F.);
#6253=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#6254=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#6255=PRECISION_QUALIFIER(4);
#6261=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6253,#6254,#6255)
)REPRESENTATION_ITEM('distance'));
#6262=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6220,#6261),#29);
#6263=PROPERTY_DEFINITION('',$,#6252);
#6264=PROPERTY_DEFINITION_REPRESENTATION(#6263,#6262);
#6265=DIRECTION_SHAPE_REPRESENTATION('',(#6218),#29);
#6266=PROPERTY_DEFINITION_REPRESENTATION(#6263,#6265);
#6267=SHAPE_ASPECT('','open boundary occurrence',#6216,.T.);
#6268=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#6227,#6267);
#6269=SHAPE_ASPECT('','pocket depth occurrence',#6216,.T.);
#6270=SHAPE_DEFINING_RELATIONSHIP('pocket depth','path feature component
usage',#6252,#6269);
#6271=FEATURE_COMPONENT_DEFINITION('',$);
#6272=PRODUCT_DEFINITION_SHAPE('',$,#6271);
#6273=POCKET_BOTTOM('bottom condition','planar',#6272,.F.);
#6274=DESCRIPTIVE_REPRESENTATION_ITEM('pocket bottom orientation','pocket
depth end');
#6275=PROPERTY_DEFINITION('',$,#6273);
#6276=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#6274),#29);
#6277=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6276);
#6278=DIRECTION('',(0.0,1.0,0.0));
#6279=DIRECTION_SHAPE_REPRESENTATION('floor normal',(#6278),#29);
#6280=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6279);
#6281=CARTESIAN_POINT('',(50.0,-10.0,90.0));
#6282=LOCATION_SHAPE_REPRESENTATION('floor location',(#6281),#29);
#6283=PROPERTY_DEFINITION_REPRESENTATION(#6275,#6282);
#6284=SHAPE_ASPECT('','bottom condition occurrence',#6216,.T.);
#6285=FEATURE_COMPONENT_RELATIONSHIP('','pocket bottom usage',#6273,#6284);
#6286=DIMENSIONAL_LOCATION('',$,#75,#2111);
#6287=PRECISION_QUALIFIER(4);
#6293=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(81.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6287)REPRESENTAT
ION_ITEM(''));
#6294=SHAPE_DIMENSION_REPRESENTATION('',(#6293),#29);
#6295=PRECISION_QUALIFIER(4);
#6296=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);

```

```

#6297=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#6298=MEASURE_QUALIFICATION('','',#6296,(#6295));
#6299=MEASURE_QUALIFICATION('','',#6297,(#6295));
#6300=TOLERANCE_VALUE(#6297,#6296);
#6301=PLUS_MINUS_TOLERANCE(#6300,#6286);
#6302=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6286,#6294);
#6308=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('',$,#38,.T.)STEP());
#6309=PROPERTY_DEFINITION('',$,#6308);
#6310=PRODUCT_DEFINITION_SHAPE('',$,#6308);
#6311=CARTESIAN_POINT('',(228.0,-40.0,180.0));
#6312=DIRECTION('',(0.0,0.0,-1.0));
#6313=DIRECTION('',(0.707106781186547,-0.707106781186547,0.0));
#6314=AXIS2_PLACEMENT_3D('orientation',#6311,#6312,#6313);
#6315=REPRESENTATION('',(1796,#2134,#2153),#29);
#6316=PROPERTY_DEFINITION_REPRESENTATION(#6309,#6315);
#6317=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(6314),#29);
#6318=PROPERTY_DEFINITION_REPRESENTATION(#6309,#6317);
#6319=FEATURE_COMPONENT_DEFINITION('',$);
#6320=PRODUCT_DEFINITION_SHAPE('',$,#6319);
#6321=PATH_FEATURE_COMPONENT('linear path','linear',#6320,.F.);
#6322=STANDARD_UNCERTAINTY('upper limit','plus',0.3000000000000000);
#6323=STANDARD_UNCERTAINTY('lower limit','minus',-0.3000000000000000);
#6324=PRECISION_QUALIFIER(4);
#6330=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(180.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6322,#6323,#6324
))REPRESENTATION_ITEM('distance'));
#6331=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(6314,#6330),#29);
#6332=PROPERTY_DEFINITION('',$,#6321);
#6333=PROPERTY_DEFINITION_REPRESENTATION(#6332,#6331);
#6334=DIRECTION_SHAPE_REPRESENTATION('',(6312),#29);
#6335=PROPERTY_DEFINITION_REPRESENTATION(#6332,#6334);
#6336=FEATURE_COMPONENT_DEFINITION('',$);
#6337=PRODUCT_DEFINITION_SHAPE('',$,#6336);
#6338=VEE_PROFILE('vee profile','v-shape',#6337,.F.);
#6343=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('profile angle'));
#6348=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-
45.0),#19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('tilt angle'));
#6349=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(6314,#6343,#6348),#29);
#6350=PROPERTY_DEFINITION('',$,#6338);
#6351=PROPERTY_DEFINITION_REPRESENTATION(#6350,#6349);
#6352=SHAPE_ASPECT('','removal boundary occurrence',#6310,.T.);
#6353=SHAPE_DEFINING_RELATIONSHIP('vee boundary','profile
usage',#6338,#6352);
#6354=SHAPE_ASPECT('','course of travel occurrence',#6310,.T.);
#6355=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#6321,#6354);
#6356=CARTESIAN_POINT('',(228.0,-40.0,180.0));
#6357=DIRECTION('',(0.0,1.0,0.0));
#6358=DIRECTION('',(1.0,0.0,0.0));
#6359=AXIS2_PLACEMENT_3D('',$,#6356,#6357,#6358);
#6360=SHAPE_ASPECT('',$,#38,.F.);

```

```

#6361=PROPERTY_DEFINITION('', $, #6360);
#6362=REPRESENTATION('', (#6359), #29);
#6363=PROPERTY_DEFINITION_REPRESENTATION(#6361, #6362);
#6364=DIMENSIONAL_LOCATION('', $, #335, #6360);
#6365=PRECISION_QUALIFIER(4);
#6371=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6365)) REPRESENTATION_ITEM(''));
#6372=SHAPE_DIMENSION_REPRESENTATION('', (#6371), #29);
#6373=PRECISION_QUALIFIER(4);
#6374=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#6375=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#6376=MEASURE_QUALIFICATION('', '#6374', (#6373));
#6377=MEASURE_QUALIFICATION('', '#6375', (#6373));
#6378=TOLERANCE_VALUE(#6375, #6374);
#6379=PLUS_MINUS_TOLERANCE(#6378, #6364);
#6380=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6364, #6372);
#6381=DIMENSIONAL_LOCATION('', $, #239, #2135);
#6382=PRECISION_QUALIFIER(4);
#6388=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(228.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6382)) REPRESENTATION_ITEM(''));
#6389=SHAPE_DIMENSION_REPRESENTATION('', (#6388), #29);
#6390=PRECISION_QUALIFIER(4);
#6391=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#6392=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#6393=MEASURE_QUALIFICATION('', '#6391', (#6390));
#6394=MEASURE_QUALIFICATION('', '#6392', (#6390));
#6395=TOLERANCE_VALUE(#6392, #6391);
#6396=PLUS_MINUS_TOLERANCE(#6395, #6381);
#6397=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6381, #6389);
#6403=(CHARACTERIZED_OBJECT('V_LIN', $) FEATURE_DEFINITION() INSTANCED_FEATURE() REMOVAL_VOLUME() SHAPE_ASPECT('V_LIN', $, #38, .T.));
#6404=PROPERTY_DEFINITION('', $, #6403);
#6405=PRODUCT_DEFINITION_SHAPE('', $, #6403);
#6406=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#6407=DIRECTION('', (0.0, 0.0, 1.0));
#6408=DIRECTION('', (1.0, 0.0, 0.0));
#6409=AXIS2_PLACEMENT_3D('orientation', #6406, #6407, #6408);
#6410=REPRESENTATION('', (#1796, #2177, #2196), #29);
#6411=PROPERTY_DEFINITION_REPRESENTATION(#6404, #6410);
#6412=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6409), #29);
#6413=PROPERTY_DEFINITION_REPRESENTATION(#6404, #6412);
#6414=SHAPE_ASPECT('', 'volume shape', #38, .T.);
#6415=PROPERTY_DEFINITION('', $, #6414);
#6416=PROPERTY_DEFINITION_REPRESENTATION(#6415, #6410);
#6417=SHAPE_ASPECT('', 'shape volume occurrence', #6405, .T.);
#6418=SHAPE_DEFINING_RELATIONSHIP('', 'volume shape usage', #6414, #6417);
#6424=(CHARACTERIZED_OBJECT('V', $) FEATURE_DEFINITION() INSTANCED_FEATURE() SHAPE_ASPECT('', $, #38, .T.) STEP());
#6425=PROPERTY_DEFINITION('', $, #6424);
#6426=PRODUCT_DEFINITION_SHAPE('', $, #6424);
#6427=CARTESIAN_POINT('', (2.0, -40.0, 178.0));

```

```

#6428=DIRECTION('', (1.0, 0.0, 0.0));
#6429=DIRECTION('', (0.0, -0.707106781186547, 0.707106781186547));
#6430=AXIS2_PLACEMENT_3D('orientation', #6427, #6428, #6429);
#6431=REPRESENTATION('', (#1796, #2220, #2239), #29);
#6432=PROPERTY_DEFINITION_REPRESENTATION(#6425, #6431);
#6433=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6430), #29);
#6434=PROPERTY_DEFINITION_REPRESENTATION(#6425, #6433);
#6435=FEATURE_COMPONENT_DEFINITION('', $);
#6436=PRODUCT_DEFINITION_SHAPE('', $, #6435);
#6437=PATH_FEATURE_COMPONENT('linear path', 'linear', #6436, .F.);
#6438=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.100000000000000);
#6439=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.100000000000000);
#6440=PRECISION_QUALIFIER(4);
#6446=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(230.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6438, #6439, #6440)) REPRESENTATION_ITEM('distance'));
#6447=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6430, #6446), #29);
#6448=PROPERTY_DEFINITION('', $, #6437);
#6449=PROPERTY_DEFINITION_REPRESENTATION(#6448, #6447);
#6450=DIRECTION_SHAPE_REPRESENTATION('', (#6428), #29);
#6451=PROPERTY_DEFINITION_REPRESENTATION(#6448, #6450);
#6452=FEATURE_COMPONENT_DEFINITION('', $);
#6453=PRODUCT_DEFINITION_SHAPE('', $, #6452);
#6454=VEE_PROFILE('vee profile', 'v-shape', #6453, .F.);
#6459=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('profile angle'));
#6464=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(-45.0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('tilt angle'));
#6465=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6430, #6459, #6464), #29);
#6466=PROPERTY_DEFINITION('', $, #6454);
#6467=PROPERTY_DEFINITION_REPRESENTATION(#6466, #6465);
#6468=SHAPE_ASPECT('', 'removal boundary occurrence', #6426, .T.);
#6469=SHAPE_DEFINING_RELATIONSHIP('vee boundary', 'profile usage', #6454, #6468);
#6470=SHAPE_ASPECT('', 'course of travel occurrence', #6426, .T.);
#6471=SHAPE_DEFINING_RELATIONSHIP('course of travel', 'path feature component usage', #6437, #6470);
#6472=CARTESIAN_POINT('', (0.0, -40.0, 178.0));
#6473=DIRECTION('', (0.0, 1.0, 0.0));
#6474=DIRECTION('', (0.0, 0.0, 1.0));
#6475=AXIS2_PLACEMENT_3D('', #6472, #6473, #6474);
#6476=SHAPE_ASPECT('', $, #38, .F.);
#6477=PROPERTY_DEFINITION('', $, #6476);
#6478=REPRESENTATION('', (#6475), #29);
#6479=PROPERTY_DEFINITION_REPRESENTATION(#6477, #6478);
#6480=DIMENSIONAL_LOCATION('', $, #335, #6476);
#6481=PRECISION_QUALIFIER(4);
#6487=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6481)) REPRESENTATION_ITEM(''));
#6488=SHAPE_DIMENSION_REPRESENTATION('', (#6487), #29);
#6489=PRECISION_QUALIFIER(4);
#6490=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000), #10);

```

```

#6491=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#6492=MEASURE_QUALIFICATION('','',#6490,(#6489));
#6493=MEASURE_QUALIFICATION('','',#6491,(#6489));
#6494=TOLERANCE_VALUE(#6491,#6490);
#6495=PLUS_MINUS_TOLERANCE(#6494,#6480);
#6496=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6480,#6488);
#6497=DIMENSIONAL_LOCATION('',$,#75,#2221);
#6498=PRECISION_QUALIFIER(4);
#6504=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(178.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6498))REPRESENTA
TION_ITEM(''));
#6505=SHAPE_DIMENSION_REPRESENTATION('',( #6504),#29);
#6506=PRECISION_QUALIFIER(4);
#6507=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000),#10);
#6508=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.3000000000000000),#10);
#6509=MEASURE_QUALIFICATION('','',#6507,(#6506));
#6510=MEASURE_QUALIFICATION('','',#6508,(#6506));
#6511=TOLERANCE_VALUE(#6508,#6507);
#6512=PLUS_MINUS_TOLERANCE(#6511,#6497);
#6513=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6497,#6505);
#6519=(CHARACTERIZED_OBJECT('V_LIN',$)FEATURE_DEFINITION()INSTANCED_FEATURE()
REMOVAL_VOLUME()SHAPE_ASPECT('V_LIN',$,#38,.T.));
#6520=PROPERTY_DEFINITION('',$,#6519);
#6521=PRODUCT_DEFINITION_SHAPE('',$,#6519);
#6522=CARTESIAN_POINT('',(0.0,0.0,0.0));
#6523=DIRECTION('',(0.0,0.0,1.0));
#6524=DIRECTION('',(1.0,0.0,0.0));
#6525=AXIS2_PLACEMENT_3D('orientation',#6522,#6523,#6524);
#6526=REPRESENTATION('',( #1796,#2263,#2282),#29);
#6527=PROPERTY_DEFINITION_REPRESENTATION(#6520,#6526);
#6528=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #6525),#29);
#6529=PROPERTY_DEFINITION_REPRESENTATION(#6520,#6528);
#6530=SHAPE_ASPECT('','volume shape',#38,.T.);
#6531=PROPERTY_DEFINITION('',$,#6530);
#6532=PROPERTY_DEFINITION_REPRESENTATION(#6531,#6526);
#6533=SHAPE_ASPECT('','shape volume occurrence',#6521,.T.);
#6534=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#6530,#6533);
#6540=(CHARACTERIZED_OBJECT('','groove')FEATURE_DEFINITION()INSTANCED_FEATURE
()REVOLVED_PROFILE()SHAPE_ASPECT('','groove',#38,.T.));
#6541=PROPERTY_DEFINITION('',$,#6540);
#6542=PRODUCT_DEFINITION_SHAPE('',$,#6540);
#6543=CARTESIAN_POINT('',(160.0,-20.0,90.0));
#6544=DIRECTION('',(0.0,-1.0,0.0));
#6545=DIRECTION('',(1.0,0.0,0.0));
#6546=AXIS2_PLACEMENT_3D('orientation',#6543,#6544,#6545);
#6547=STANDARD_UNCERTAINTY('upper limit','plus',0.0500000000000000);
#6548=STANDARD_UNCERTAINTY('lower limit','minus',0.0);
#6549=PRECISION_QUALIFIER(4);
#6555=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(22.0),#10)QUALIFIED_REPRESENTATION_ITEM((#6547,#6548,#6549)
)REPRESENTATION_ITEM('radius'));
#6556=REPRESENTATION('',( #2316,#2364,#2388,#2404),#29);
#6557=PROPERTY_DEFINITION_REPRESENTATION(#6541,#6556);

```



```

#6558=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6546, #6555), #29);
#6559=PROPERTY_DEFINITION_REPRESENTATION(#6541, #6558);
#6560=DIRECTION('', (-1.0, 0.0, 0.0));
#6561=DIRECTION_SHAPE_REPRESENTATION('removal direction', (#6560), #29);
#6562=PROPERTY_DEFINITION_REPRESENTATION(#6541, #6561);
#6563=CARTESIAN_POINT('', (182.0, -20.0, 90.0));
#6564=DIRECTION('', (0.0, 0.0, -1.0));
#6565=DIRECTION('', (0.0, -1.0, 0.0));
#6566=AXIS2_PLACEMENT_3D('orientation', #6563, #6564, #6565);
#6567=FEATURE_COMPONENT_DEFINITION('', $);
#6568=PRODUCT_DEFINITION_SHAPE('', $, #6567);
#6569=SQUARE_U_PROFILE('', 'sweep', #6568, .F.);
#6570=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#6571=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#6572=PRECISION_QUALIFIER(4);
#6578=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(3.5000000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#6570
, #6571, #6572))REPRESENTATION_ITEM('width'));
#6583=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#6588=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#6589=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6566, #6578, #6583, #6588), #29);
#6590=PROPERTY_DEFINITION('', $, #6569);
#6591=PROPERTY_DEFINITION_REPRESENTATION(#6590, #6589);
#6592=SHAPE_ASPECT('', 'sweep occurrence', #6542, .T.);
#6593=SHAPE_DEFINING_RELATIONSHIP('', 'profile usage', #6569, #6592);
#6594=DIMENSIONAL_LOCATION('', $, #1110, #2317);
#6595=PRECISION_QUALIFIER(4);
#6601=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.2500000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#659
5))REPRESENTATION_ITEM(''));
#6602=SHAPE_DIMENSION_REPRESENTATION('', (#6601), #29);
#6603=PRECISION_QUALIFIER(4);
#6604=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0500000000000000), #10);
#6605=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0500000000000000), #10);
#6606=MEASURE_QUALIFICATION('', '#6604', (#6603));
#6607=MEASURE_QUALIFICATION('', '#6605', (#6603));
#6608=TOLERANCE_VALUE(#6605, #6604);
#6609=PLUS_MINUS_TOLERANCE(#6608, #6594);
#6610=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6594, #6602);
#6617=(CHARACTERIZED_OBJECT('compound feature in
solid', 'counterbore')COMPOSITE_HOLE()
  COMPOUND_FEATURE()FEATURE_DEFINITION()INSTANCED_FEATURE()
  SHAPE_ASPECT('compound feature in solid', 'counterbore', #38, .T.));
#6618=PROPERTY_DEFINITION('', $, #6617);
#6619=PRODUCT_DEFINITION_SHAPE('', $, #6617);
#6620=CARTESIAN_POINT('', (160.0, 0.0, 30.0));
#6621=DIRECTION('', (0.0, -1.0, 0.0));
#6622=DIRECTION('', (-1.0, 0.0, 0.0));
#6623=AXIS2_PLACEMENT_3D('orientation', #6620, #6621, #6622);
#6624=REPRESENTATION('', (#2434, #2468, #2508, #2532, #2548, #2564), #29);
#6625=PROPERTY_DEFINITION_REPRESENTATION(#6618, #6624);

```

```

#6626=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6623), #29);
#6627=PROPERTY_DEFINITION_REPRESENTATION(#6618, #6626);

#6633=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#6634=PROPERTY_DEFINITION('', $, #6633);
#6635=PRODUCT_DEFINITION_SHAPE('', $, #6633);
#6636=CARTESIAN_POINT('', (160.0, 0.0, 30.0));
#6637=DIRECTION('', (0.0, -1.0, 0.0));
#6638=DIRECTION('', (-1.0, 0.0, 0.0));
#6639=AXIS2_PLACEMENT_3D('orientation', #6636, #6637, #6638);
#6640=REPRESENTATION('', (#2434, #2468, #2548), #29);
#6641=PROPERTY_DEFINITION_REPRESENTATION(#6634, #6640);
#6642=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6639), #29);
#6643=PROPERTY_DEFINITION_REPRESENTATION(#6634, #6642);
#6644=FEATURE_COMPONENT_DEFINITION('', $);
#6645=PRODUCT_DEFINITION_SHAPE('', $, #6644);
#6646=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6645, .F.);
#6647=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#6648=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#6649=PRECISION_QUALIFIER(4);
#6655=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0), #10)

QUALIFIED_REPRESENTATION_ITEM((#6647, #6648, #6649)) REPRESENTATION_ITEM('diamet
er'));
#6656=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6639, #6655), #29);
#6657=PROPERTY_DEFINITION('', $, #6646);
#6658=PROPERTY_DEFINITION_REPRESENTATION(#6657, #6656);
#6659=FEATURE_COMPONENT_DEFINITION('', $);
#6660=PRODUCT_DEFINITION_SHAPE('', $, #6659);
#6661=PATH_FEATURE_COMPONENT('linear path', 'linear', #6660, .F.);
#6662=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#6663=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#6664=PRECISION_QUALIFIER(4);
#6670=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0), #10)

QUALIFIED_REPRESENTATION_ITEM((#6662, #6663, #6664)) REPRESENTATION_ITEM('distan
ce'));
#6671=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6639, #6670), #29);
#6672=PROPERTY_DEFINITION('', $, #6661);
#6673=PROPERTY_DEFINITION_REPRESENTATION(#6672, #6671);
#6674=DIRECTION_SHAPE_REPRESENTATION('', (#6637), #29);
#6675=PROPERTY_DEFINITION_REPRESENTATION(#6672, #6674);
#6676=SHAPE_ASPECT('', 'diameter occurrence', #6635, .T.);
#6677=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6646, #6676);
#6678=SHAPE_ASPECT('', 'hole depth occurrence', #6635, .T.);
#6679=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6661, #6678);

#6680=FEATURE_COMPONENT_DEFINITION('', $);
#6681=PRODUCT_DEFINITION_SHAPE('', $, #6680);

```

```

#6682=HOLE_BOTTOM('bottom condition','flat',#6681,.F.);
#6687=SHAPE_ASPECT('', 'bottom condition occurrence',#6635,.T.);
#6688=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage', #6682, #6687);

#6689=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #6619,.T.);
#6695=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38,.T.));
#6696=PROPERTY_DEFINITION('', $, #6695);
#6697=PRODUCT_DEFINITION_SHAPE('', $, #6695);
#6698=CARTESIAN_POINT('', (160.0, 0.0, 30.0));
#6699=DIRECTION('', (0.0, -1.0, 0.0));
#6700=DIRECTION('', (-1.0, 0.0, 0.0));
#6701=AXIS2_PLACEMENT_3D('orientation', #6698, #6699, #6700);
#6702=REPRESENTATION('', (#2508, #2532, #2564), #29);
#6703=PROPERTY_DEFINITION_REPRESENTATION(#6696, #6702);
#6704=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6701), #29);
#6705=PROPERTY_DEFINITION_REPRESENTATION(#6696, #6704);
#6706=FEATURE_COMPONENT_DEFINITION('', $);
#6707=PRODUCT_DEFINITION_SHAPE('', $, #6706);
#6708=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6707,.F.);
#6709=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.5000000000000000);
#6710=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.5000000000000000);
#6711=PRECISION_QUALIFIER(4);
#6717=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6709, #6710, #6711)
) REPRESENTATION_ITEM('diameter'));
#6718=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6701, #6717), #29);
#6719=PROPERTY_DEFINITION('', $, #6708);
#6720=PROPERTY_DEFINITION_REPRESENTATION(#6719, #6718);
#6721=FEATURE_COMPONENT_DEFINITION('', $);
#6722=PRODUCT_DEFINITION_SHAPE('', $, #6721);
#6723=PATH_FEATURE_COMPONENT('linear path', 'linear', #6722,.F.);
#6724=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#6725=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#6726=PRECISION_QUALIFIER(4);
#6732=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6724, #6725, #6726)
) REPRESENTATION_ITEM('distance'));
#6733=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6701, #6732), #29);
#6734=PROPERTY_DEFINITION('', $, #6723);
#6735=PROPERTY_DEFINITION_REPRESENTATION(#6734, #6733);
#6736=DIRECTION_SHAPE_REPRESENTATION('', (#6699), #29);
#6737=PROPERTY_DEFINITION_REPRESENTATION(#6734, #6736);
#6738=SHAPE_ASPECT('', 'diameter occurrence', #6697,.T.);
#6739=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6708, #6738);
#6740=SHAPE_ASPECT('', 'hole depth occurrence', #6697,.T.);
#6741=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6723, #6740);

#6742=FEATURE_COMPONENT_DEFINITION('', $);
#6743=PRODUCT_DEFINITION_SHAPE('', $, #6742);
#6744=HOLE_BOTTOM('bottom condition', 'flat', #6743,.F.);

```

```

#6749=SHAPE_ASPECT('', 'bottom condition occurrence', #6697, .T.);

#6750=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #6744, #6749);
#6751=FEATURE_COMPONENT_RELATIONSHIP('large hole', $, #6689, #6695);
#6752=FEATURE_COMPONENT_RELATIONSHIP('small hole', $, #6689, #6633);
#6753=CARTESIAN_POINT('', (160.0, -30.0, 30.0));
#6754=DIRECTION('', (0.0, 1.0, 0.0));
#6755=AXIS1_PLACEMENT('', #6753, #6754);
#6756=SHAPE_ASPECT('', $, #38, .F.);
#6757=PROPERTY_DEFINITION('', $, #6756);
#6758=REPRESENTATION('', (#6755), #29);
#6759=PROPERTY_DEFINITION_REPRESENTATION(#6757, #6758);
#6760=DIMENSIONAL_LOCATION('', $, #1261, #6756);
#6761=PRECISION_QUALIFIER(4);
#6767=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0), #10)QUALIFIED_REPRESENTATION_ITEM((#6761))REPRESENTAT
ION_ITEM(''));
#6768=SHAPE_DIMENSION_REPRESENTATION('', (#6767), #29);
#6769=PRECISION_QUALIFIER(4);
#6770=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#6771=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000), #10);
#6772=MEASURE_QUALIFICATION('', '#6770', (#6769));
#6773=MEASURE_QUALIFICATION('', '#6771', (#6769));
#6774=TOLERANCE_VALUE(#6771, #6770);
#6775=PLUS_MINUS_TOLERANCE(#6774, #6760);
#6776=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6760, #6768);
#6777=CARTESIAN_POINT('', (160.0, -30.0, 30.0));
#6778=DIRECTION('', (0.0, 1.0, 0.0));
#6779=AXIS1_PLACEMENT('', #6777, #6778);
#6780=SHAPE_ASPECT('', $, #38, .F.);
#6781=PROPERTY_DEFINITION('', $, #6780);
#6782=REPRESENTATION('', (#6779), #29);
#6783=PROPERTY_DEFINITION_REPRESENTATION(#6781, #6782);
#6784=DIMENSIONAL_LOCATION('', $, #75, #6780);
#6785=PRECISION_QUALIFIER(4);
#6791=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0), #10)QUALIFIED_REPRESENTATION_ITEM((#6785))REPRESENTAT
ION_ITEM(''));
#6792=SHAPE_DIMENSION_REPRESENTATION('', (#6791), #29);
#6793=PRECISION_QUALIFIER(4);
#6794=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#6795=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000), #10);
#6796=MEASURE_QUALIFICATION('', '#6794', (#6793));
#6797=MEASURE_QUALIFICATION('', '#6795', (#6793));
#6798=TOLERANCE_VALUE(#6795, #6794);
#6799=PLUS_MINUS_TOLERANCE(#6798, #6784);
#6800=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6784, #6792);
#6807=(CHARACTERIZED_OBJECT('compound feature in
solid', 'counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid', 'counterbore', #38, .T.));
#6808=PROPERTY_DEFINITION('', $, #6807);

```

```

#6809=PRODUCT_DEFINITION_SHAPE('', $, #6807);
#6810=CARTESIAN_POINT('', (160.0, 0.0, 150.0));
#6811=DIRECTION('', (0.0, -1.0, 0.0));
#6812=DIRECTION('', (-1.0, 0.0, 0.0));
#6813=AXIS2_PLACEMENT_3D('orientation', #6810, #6811, #6812);
#6814=REPRESENTATION('', (#2598, #2638, #2662, #2678, #2694), #29);
#6815=PROPERTY_DEFINITION_REPRESENTATION(#6808, #6814);
#6816=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6813), #29);
#6817=PROPERTY_DEFINITION_REPRESENTATION(#6808, #6816);
#6823=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#6824=PROPERTY_DEFINITION('', $, #6823);
#6825=PRODUCT_DEFINITION_SHAPE('', $, #6823);
#6826=CARTESIAN_POINT('', (160.0, 0.0, 150.0));
#6827=DIRECTION('', (0.0, -1.0, 0.0));
#6828=DIRECTION('', (-1.0, 0.0, 0.0));
#6829=AXIS2_PLACEMENT_3D('orientation', #6826, #6827, #6828);
#6830=REPRESENTATION('', (#2598, #2678), #29);
#6831=PROPERTY_DEFINITION_REPRESENTATION(#6824, #6830);
#6832=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6829), #29);
#6833=PROPERTY_DEFINITION_REPRESENTATION(#6824, #6832);
#6834=FEATURE_COMPONENT_DEFINITION('', $);
#6835=PRODUCT_DEFINITION_SHAPE('', $, #6834);
#6836=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6835, .F.);
#6837=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#6838=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#6839=PRECISION_QUALIFIER(4);
#6845=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6837, #6838, #6839)
) REPRESENTATION_ITEM('diameter'));
#6846=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6829, #6845), #29);
#6847=PROPERTY_DEFINITION('', $, #6836);
#6848=PROPERTY_DEFINITION_REPRESENTATION(#6847, #6846);
#6849=FEATURE_COMPONENT_DEFINITION('', $);
#6850=PRODUCT_DEFINITION_SHAPE('', $, #6849);
#6851=PATH_FEATURE_COMPONENT('linear path', 'linear', #6850, .F.);
#6852=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#6853=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#6854=PRECISION_QUALIFIER(4);
#6860=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6852, #6853, #6854)
) REPRESENTATION_ITEM('distance'));
#6861=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6829, #6860), #29);
#6862=PROPERTY_DEFINITION('', $, #6851);
#6863=PROPERTY_DEFINITION_REPRESENTATION(#6862, #6861);
#6864=DIRECTION_SHAPE_REPRESENTATION('', (#6827), #29);
#6865=PROPERTY_DEFINITION_REPRESENTATION(#6862, #6864);
#6866=SHAPE_ASPECT('', 'diameter occurrence', #6825, .T.);
#6867=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6836, #6866);
#6868=SHAPE_ASPECT('', 'hole depth occurrence', #6825, .T.);
#6869=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6851, #6868);

```

```

#6870=FEATURE_COMPONENT_DEFINITION('', $);
#6871=PRODUCT_DEFINITION_SHAPE('', $, #6870);
#6872=HOLE_BOTTOM('bottom condition', 'flat', #6871, .F.);
#6877=SHAPE_ASPECT('', 'bottom condition occurrence', #6825, .T.);

#6878=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #6872, #6877);
#6879=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #6809, .T.);
#6885=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#6886=PROPERTY_DEFINITION('', $, #6885);
#6887=PRODUCT_DEFINITION_SHAPE('', $, #6885);
#6888=CARTESIAN_POINT('', (160.0, 0.0, 150.0));
#6889=DIRECTION('', (0.0, -1.0, 0.0));
#6890=DIRECTION('', (-1.0, 0.0, 0.0));
#6891=AXIS2_PLACEMENT_3D('orientation', #6888, #6889, #6890);
#6892=REPRESENTATION('', (#2638, #2662, #2694), #29);
#6893=PROPERTY_DEFINITION_REPRESENTATION(#6886, #6892);
#6894=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6891), #29);
#6895=PROPERTY_DEFINITION_REPRESENTATION(#6886, #6894);
#6896=FEATURE_COMPONENT_DEFINITION('', $);
#6897=PRODUCT_DEFINITION_SHAPE('', $, #6896);
#6898=CIRCULAR_CLOSED_PROFILE('circular profile', $, #6897, .F.);
#6899=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.5000000000000000);
#6900=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.5000000000000000);
#6901=PRECISION_QUALIFIER(4);
#6907=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6899, #6900, #6901)
) REPRESENTATION_ITEM('diameter'));
#6908=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6891, #6907), #29);
#6909=PROPERTY_DEFINITION('', $, #6898);
#6910=PROPERTY_DEFINITION_REPRESENTATION(#6909, #6908);
#6911=FEATURE_COMPONENT_DEFINITION('', $);
#6912=PRODUCT_DEFINITION_SHAPE('', $, #6911);
#6913=PATH_FEATURE_COMPONENT('linear path', 'linear', #6912, .F.);
#6914=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#6915=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#6916=PRECISION_QUALIFIER(4);
#6922=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6914, #6915, #6916)
) REPRESENTATION_ITEM('distance'));
#6923=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#6891, #6922), #29);
#6924=PROPERTY_DEFINITION('', $, #6913);
#6925=PROPERTY_DEFINITION_REPRESENTATION(#6924, #6923);
#6926=DIRECTION_SHAPE_REPRESENTATION('', (#6889), #29);
#6927=PROPERTY_DEFINITION_REPRESENTATION(#6924, #6926);
#6928=SHAPE_ASPECT('', 'diameter occurrence', #6887, .T.);
#6929=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #6898, #6928);
#6930=SHAPE_ASPECT('', 'hole depth occurrence', #6887, .T.);
#6931=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #6913, #6930);

#6932=FEATURE_COMPONENT_DEFINITION('', $);

```

```

#6933=PRODUCT_DEFINITION_SHAPE('', $, #6932);
#6934=HOLE_BOTTOM('bottom condition', 'flat', #6933, .F.);
#6939=SHAPE_ASPECT('', 'bottom condition occurrence', #6887, .T.);

#6940=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #6934, #6939);
#6941=FEATURE_COMPONENT_RELATIONSHIP('large hole', $, #6879, #6885);
#6942=FEATURE_COMPONENT_RELATIONSHIP('small hole', $, #6879, #6823);
#6943=CARTESIAN_POINT('', (160.0, -40.0, 150.0));
#6944=DIRECTION('', (0.0, 1.0, 0.0));
#6945=AXIS1_PLACEMENT('', #6943, #6944);
#6946=SHAPE_ASPECT('', $, #38, .F.);
#6947=PROPERTY_DEFINITION('', $, #6946);
#6948=REPRESENTATION('', (#6945), #29);
#6949=PROPERTY_DEFINITION_REPRESENTATION(#6947, #6948);
#6950=DIMENSIONAL_LOCATION('', $, #1261, #6946);
#6951=PRECISION_QUALIFIER(4);
#6957=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(70.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6951)) REPRESENTAT
ION_ITEM(''));
#6958=SHAPE_DIMENSION_REPRESENTATION('', (#6957), #29);
#6959=PRECISION_QUALIFIER(4);
#6960=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#6961=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000), #10);
#6962=MEASURE_QUALIFICATION('', '#6960, (#6959)');
#6963=MEASURE_QUALIFICATION('', '#6961, (#6959)');
#6964=TOLERANCE_VALUE(#6961, #6960);
#6965=PLUS_MINUS_TOLERANCE(#6964, #6950);
#6966=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6950, #6958);
#6967=CARTESIAN_POINT('', (160.0, -40.0, 150.0));
#6968=DIRECTION('', (0.0, 1.0, 0.0));
#6969=AXIS1_PLACEMENT('', #6967, #6968);
#6970=SHAPE_ASPECT('', $, #38, .F.);
#6971=PROPERTY_DEFINITION('', $, #6970);
#6972=REPRESENTATION('', (#6969), #29);
#6973=PROPERTY_DEFINITION_REPRESENTATION(#6971, #6972);
#6974=DIMENSIONAL_LOCATION('', $, #143, #6970);
#6975=PRECISION_QUALIFIER(4);
#6981=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0), #10) QUALIFIED_REPRESENTATION_ITEM((#6975)) REPRESENTAT
ION_ITEM(''));
#6982=SHAPE_DIMENSION_REPRESENTATION('', (#6981), #29);
#6983=PRECISION_QUALIFIER(4);
#6984=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#6985=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000), #10);
#6986=MEASURE_QUALIFICATION('', '#6984, (#6983)');
#6987=MEASURE_QUALIFICATION('', '#6985, (#6983)');
#6988=TOLERANCE_VALUE(#6985, #6984);
#6989=PLUS_MINUS_TOLERANCE(#6988, #6974);
#6990=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#6974, #6982);
#6997=(CHARACTERIZED_OBJECT('compound feature in
solid', 'counterbore') COMPOSITE_HOLE() COMPOUND_FEATURE() FEATURE_DEFINITION() IN

```

```

STANCED_FEATURE() SHAPE_ASPECT('compound feature in
solid', 'counterbore', #38, .T.);
#6998=PROPERTY_DEFINITION('', $, #6997);
#6999=PRODUCT_DEFINITION_SHAPE('', $, #6997);
#7000=CARTESIAN_POINT('', (205.0, 0.0, 27.500000000000000));
#7001=DIRECTION('', (0.0, -1.0, 0.0));
#7002=DIRECTION('', (-1.0, 0.0, 0.0));
#7003=AXIS2_PLACEMENT_3D('orientation', #7000, #7001, #7002);
#7004=REPRESENTATION('', (#2728, #2768, #2792, #2808, #2824), #29);
#7005=PROPERTY_DEFINITION_REPRESENTATION(#6998, #7004);
#7006=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7003), #29);
#7007=PROPERTY_DEFINITION_REPRESENTATION(#6998, #7006);
#7013=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.);
#7014=PROPERTY_DEFINITION('', $, #7013);
#7015=PRODUCT_DEFINITION_SHAPE('', $, #7013);
#7016=CARTESIAN_POINT('', (205.0, 0.0, 27.500000000000000));
#7017=DIRECTION('', (0.0, -1.0, 0.0));
#7018=DIRECTION('', (-1.0, 0.0, 0.0));
#7019=AXIS2_PLACEMENT_3D('orientation', #7016, #7017, #7018);
#7020=REPRESENTATION('', (#2728, #2808), #29);
#7021=PROPERTY_DEFINITION_REPRESENTATION(#7014, #7020);
#7022=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7019), #29);
#7023=PROPERTY_DEFINITION_REPRESENTATION(#7014, #7022);
#7024=FEATURE_COMPONENT_DEFINITION('', $);
#7025=PRODUCT_DEFINITION_SHAPE('', $, #7024);
#7026=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7025, .F.);
#7027=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.250000000000000);
#7028=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.250000000000000);
#7029=PRECISION_QUALIFIER(4);
#7035=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7027, #7028, #7029)
) REPRESENTATION_ITEM('diameter'));
#7036=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7019, #7035), #29);
#7037=PROPERTY_DEFINITION('', $, #7026);
#7038=PROPERTY_DEFINITION_REPRESENTATION(#7037, #7036);
#7039=FEATURE_COMPONENT_DEFINITION('', $);
#7040=PRODUCT_DEFINITION_SHAPE('', $, #7039);
#7041=PATH_FEATURE_COMPONENT('linear path', 'linear', #7040, .F.);
#7042=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.100000000000000);
#7043=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.100000000000000);
#7044=PRECISION_QUALIFIER(4);
#7050=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7042, #7043, #7044)
) REPRESENTATION_ITEM('distance'));
#7051=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7019, #7050), #29);
#7052=PROPERTY_DEFINITION('', $, #7041);
#7053=PROPERTY_DEFINITION_REPRESENTATION(#7052, #7051);
#7054=DIRECTION_SHAPE_REPRESENTATION('', (#7017), #29);
#7055=PROPERTY_DEFINITION_REPRESENTATION(#7052, #7054);
#7056=SHAPE_ASPECT('', 'diameter occurrence', #7015, .T.);
#7057=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7026, #7056);
#7058=SHAPE_ASPECT('', 'hole depth occurrence', #7015, .T.);

```



```

#7059=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage', #7041, #7058);

#7060=FEATURE_COMPONENT_DEFINITION('', $);
#7061=PRODUCT_DEFINITION_SHAPE('', $, #7060);
#7062=HOLE_BOTTOM('bottom condition', 'flat', #7061, .F.);
#7067=SHAPE_ASPECT('', 'bottom condition occurrence', #7015, .T.);

#7068=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7062, #7067);
#7069=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #6999, .T.);
#7075=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#7076=PROPERTY_DEFINITION('', $, #7075);
#7077=PRODUCT_DEFINITION_SHAPE('', $, #7075);
#7078=CARTESIAN_POINT('', (205.0, 0.0, 27.500000000000000));
#7079=DIRECTION('', (0.0, -1.0, 0.0));
#7080=DIRECTION('', (-1.0, 0.0, 0.0));
#7081=AXIS2_PLACEMENT_3D('orientation', #7078, #7079, #7080);
#7082=REPRESENTATION('', (#2768, #2792, #2824), #29);
#7083=PROPERTY_DEFINITION_REPRESENTATION(#7076, #7082);
#7084=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7081), #29);
#7085=PROPERTY_DEFINITION_REPRESENTATION(#7076, #7084);
#7086=FEATURE_COMPONENT_DEFINITION('', $);
#7087=PRODUCT_DEFINITION_SHAPE('', $, #7086);
#7088=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7087, .F.);
#7089=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.500000000000000);
#7090=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.500000000000000);
#7091=PRECISION_QUALIFIER(4);
#7097=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7089, #7090, #7091)
) REPRESENTATION_ITEM('diameter'));
#7098=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7081, #7097), #29);
#7099=PROPERTY_DEFINITION('', $, #7088);
#7100=PROPERTY_DEFINITION_REPRESENTATION(#7099, #7098);
#7101=FEATURE_COMPONENT_DEFINITION('', $);
#7102=PRODUCT_DEFINITION_SHAPE('', $, #7101);
#7103=PATH_FEATURE_COMPONENT('linear path', 'linear', #7102, .F.);
#7104=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#7105=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#7106=PRECISION_QUALIFIER(4);
#7112=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7104, #7105, #7106)
) REPRESENTATION_ITEM('distance'));
#7113=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7081, #7112), #29);
#7114=PROPERTY_DEFINITION('', $, #7103);
#7115=PROPERTY_DEFINITION_REPRESENTATION(#7114, #7113);
#7116=DIRECTION_SHAPE_REPRESENTATION('', (#7079), #29);
#7117=PROPERTY_DEFINITION_REPRESENTATION(#7114, #7116);
#7118=SHAPE_ASPECT('', 'diameter occurrence', #7077, .T.);
#7119=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7088, #7118);
#7120=SHAPE_ASPECT('', 'hole depth occurrence', #7077, .T.);

```

ISO 10303-223:2008 (E)

```

#7121=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage', #7103, #7120);

#7122=FEATURE_COMPONENT_DEFINITION('', $);
#7123=PRODUCT_DEFINITION_SHAPE('', $, #7122);
#7124=HOLE_BOTTOM('bottom condition', 'flat', #7123, .F.);
#7129=SHAPE_ASPECT('', 'bottom condition occurrence', #7077, .T.);

#7130=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7124, #7129);
#7131=FEATURE_COMPONENT_RELATIONSHIP('large hole', $, #7069, #7075);
#7132=FEATURE_COMPONENT_RELATIONSHIP('small hole', $, #7069, #7013);
#7133=CARTESIAN_POINT('', (205.0, -50.0, 27.500000000000000));
#7134=DIRECTION('', (0.0, 1.0, 0.0));
#7135=AXIS1_PLACEMENT('', #7133, #7134);
#7136=SHAPE_ASPECT('', $, #38, .F.);
#7137=PROPERTY_DEFINITION('', $, #7136);
#7138=REPRESENTATION('', (#7135), #29);
#7139=PROPERTY_DEFINITION_REPRESENTATION(#7137, #7138);
#7140=DIMENSIONAL_LOCATION('', $, #1261, #7136);
#7141=PRECISION_QUALIFIER(4);
#7147=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7141))REPRESENTAT
ION_ITEM(''));
#7148=SHAPE_DIMENSION_REPRESENTATION('', (#7147), #29);
#7149=PRECISION_QUALIFIER(4);
#7150=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000), #10);
#7151=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000), #10);
#7152=MEASURE_QUALIFICATION('', '#7150, (#7149));
#7153=MEASURE_QUALIFICATION('', '#7151, (#7149));
#7154=TOLERANCE_VALUE(#7151, #7150);
#7155=PLUS_MINUS_TOLERANCE(#7154, #7140);
#7156=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7140, #7148);
#7157=CARTESIAN_POINT('', (205.0, -50.0, 27.500000000000000));
#7158=DIRECTION('', (0.0, 1.0, 0.0));
#7159=AXIS1_PLACEMENT('', #7157, #7158);
#7160=SHAPE_ASPECT('', $, #38, .F.);
#7161=PROPERTY_DEFINITION('', $, #7160);
#7162=REPRESENTATION('', (#7159), #29);
#7163=PROPERTY_DEFINITION_REPRESENTATION(#7161, #7162);
#7164=DIMENSIONAL_LOCATION('', $, #75, #7160);
#7165=PRECISION_QUALIFIER(4);
#7171=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#716
5))REPRESENTATION_ITEM(''));
#7172=SHAPE_DIMENSION_REPRESENTATION('', (#7171), #29);
#7173=PRECISION_QUALIFIER(4);
#7174=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000), #10);
#7175=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000), #10);
#7176=MEASURE_QUALIFICATION('', '#7174, (#7173));
#7177=MEASURE_QUALIFICATION('', '#7175, (#7173));
#7178=TOLERANCE_VALUE(#7175, #7174);
#7179=PLUS_MINUS_TOLERANCE(#7178, #7164);

```

```

#7180=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7164,#7172);
#7187=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#7188=PROPERTY_DEFINITION('',$, #7187);
#7189=PRODUCT_DEFINITION_SHAPE('',$, #7187);
#7190=CARTESIAN_POINT('', (205.0,0.0,152.50000000000000));
#7191=DIRECTION('', (0.0,-1.0,0.0));
#7192=DIRECTION('', (-1.0,0.0,0.0));
#7193=AXIS2_PLACEMENT_3D('orientation',#7190,#7191,#7192);
#7194=REPRESENTATION('', (#2858,#2898,#2922,#2938,#2954),#29);
#7195=PROPERTY_DEFINITION_REPRESENTATION(#7188,#7194);
#7196=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7193),#29);
#7197=PROPERTY_DEFINITION_REPRESENTATION(#7188,#7196);
#7203=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('', $, #38, .T.));
#7204=PROPERTY_DEFINITION('', $, #7203);
#7205=PRODUCT_DEFINITION_SHAPE('', $, #7203);
#7206=CARTESIAN_POINT('', (205.0,0.0,152.50000000000000));
#7207=DIRECTION('', (0.0,-1.0,0.0));
#7208=DIRECTION('', (-1.0,0.0,0.0));
#7209=AXIS2_PLACEMENT_3D('orientation',#7206,#7207,#7208);
#7210=REPRESENTATION('', (#2858,#2938),#29);
#7211=PROPERTY_DEFINITION_REPRESENTATION(#7204,#7210);
#7212=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7209),#29);
#7213=PROPERTY_DEFINITION_REPRESENTATION(#7204,#7212);
#7214=FEATURE_COMPONENT_DEFINITION('', $);
#7215=PRODUCT_DEFINITION_SHAPE('', $, #7214);
#7216=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7215, .F.);
#7217=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#7218=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#7219=PRECISION_QUALIFIER(4);
#7225=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7217,#7218,#7219)
)REPRESENTATION_ITEM('diameter'));
#7226=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7209,#7225),#29);
#7227=PROPERTY_DEFINITION('', $, #7216);
#7228=PROPERTY_DEFINITION_REPRESENTATION(#7227,#7226);
#7229=FEATURE_COMPONENT_DEFINITION('', $);
#7230=PRODUCT_DEFINITION_SHAPE('', $, #7229);
#7231=PATH_FEATURE_COMPONENT('linear path', 'linear', #7230, .F.);
#7232=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#7233=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#7234=PRECISION_QUALIFIER(4);
#7240=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7232,#7233,#7234)
)REPRESENTATION_ITEM('distance'));
#7241=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7209,#7240),#29);
#7242=PROPERTY_DEFINITION('', $, #7231);
#7243=PROPERTY_DEFINITION_REPRESENTATION(#7242,#7241);
#7244=DIRECTION_SHAPE_REPRESENTATION('', (#7207),#29);
#7245=PROPERTY_DEFINITION_REPRESENTATION(#7242,#7244);

```

```

#7246=SHAPE_ASPECT('', 'diameter occurrence', #7205, .T.);
#7247=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7216, #7246);
#7248=SHAPE_ASPECT('', 'hole depth occurrence', #7205, .T.);
#7249=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #7231, #7248);

#7250=FEATURE_COMPONENT_DEFINITION('', $);
#7251=PRODUCT_DEFINITION_SHAPE('', $, #7250);
#7252=HOLE_BOTTOM('bottom condition', 'flat', #7251, .F.);
#7257=SHAPE_ASPECT('', 'bottom condition occurrence', #7205, .T.);

#7258=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7252, #7257);
#7259=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #7189, .T.);
#7265=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#7266=PROPERTY_DEFINITION('', $, #7265);
#7267=PRODUCT_DEFINITION_SHAPE('', $, #7265);
#7268=CARTESIAN_POINT('', (205.0, 0.0, 152.50000000000000));
#7269=DIRECTION('', (0.0, -1.0, 0.0));
#7270=DIRECTION('', (-1.0, 0.0, 0.0));
#7271=AXIS2_PLACEMENT_3D('orientation', #7268, #7269, #7270);
#7272=REPRESENTATION('', (#2898, #2922, #2954), #29);
#7273=PROPERTY_DEFINITION_REPRESENTATION(#7266, #7272);
#7274=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7271), #29);
#7275=PROPERTY_DEFINITION_REPRESENTATION(#7266, #7274);
#7276=FEATURE_COMPONENT_DEFINITION('', $);
#7277=PRODUCT_DEFINITION_SHAPE('', $, #7276);
#7278=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7277, .F.);
#7279=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.5000000000000000);
#7280=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.5000000000000000);
#7281=PRECISION_QUALIFIER(4);
#7287=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7279, #7280, #7281)
) REPRESENTATION_ITEM('diameter'));
#7288=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7271, #7287), #29);
#7289=PROPERTY_DEFINITION('', $, #7278);
#7290=PROPERTY_DEFINITION_REPRESENTATION(#7289, #7288);
#7291=FEATURE_COMPONENT_DEFINITION('', $);
#7292=PRODUCT_DEFINITION_SHAPE('', $, #7291);
#7293=PATH_FEATURE_COMPONENT('linear path', 'linear', #7292, .F.);
#7294=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#7295=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#7296=PRECISION_QUALIFIER(4);
#7302=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7294, #7295, #7296)
) REPRESENTATION_ITEM('distance'));
#7303=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7271, #7302), #29);
#7304=PROPERTY_DEFINITION('', $, #7293);
#7305=PROPERTY_DEFINITION_REPRESENTATION(#7304, #7303);
#7306=DIRECTION_SHAPE_REPRESENTATION('', (#7269), #29);
#7307=PROPERTY_DEFINITION_REPRESENTATION(#7304, #7306);
#7308=SHAPE_ASPECT('', 'diameter occurrence', #7267, .T.);

```

```

#7309=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#7278,#7308);
#7310=SHAPE_ASPECT('', 'hole depth occurrence',#7267,.T.);
#7311=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#7293,#7310);

#7312=FEATURE_COMPONENT_DEFINITION('', $);
#7313=PRODUCT_DEFINITION_SHAPE('', $,#7312);
#7314=HOLE_BOTTOM('bottom condition','flat',#7313,.F.);
#7319=SHAPE_ASPECT('', 'bottom condition occurrence',#7267,.T.);

#7320=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#7314,#7319);
#7321=FEATURE_COMPONENT_RELATIONSHIP('large hole',$,#7259,#7265);
#7322=FEATURE_COMPONENT_RELATIONSHIP('small hole',$,#7259,#7203);
#7323=CARTESIAN_POINT('', (205.0,-50.0,152.50000000000000));
#7324=DIRECTION('', (0.0,1.0,0.0));
#7325=AXIS1_PLACEMENT('', #7323,#7324);
#7326=SHAPE_ASPECT('', $,#38,.F.);
#7327=PROPERTY_DEFINITION('', $,#7326);
#7328=REPRESENTATION('', (#7325), #29);
#7329=PROPERTY_DEFINITION_REPRESENTATION(#7327,#7328);
#7330=DIMENSIONAL_LOCATION('', $,#1261,#7326);
#7331=PRECISION_QUALIFIER(4);
#7337=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7331))REPRESENTAT
ION_ITEM(''));
#7338=SHAPE_DIMENSION_REPRESENTATION('', (#7337), #29);
#7339=PRECISION_QUALIFIER(4);
#7340=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#7341=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#7342=MEASURE_QUALIFICATION('', '#7340', (#7339));
#7343=MEASURE_QUALIFICATION('', '#7341', (#7339));
#7344=TOLERANCE_VALUE(#7341,#7340);
#7345=PLUS_MINUS_TOLERANCE(#7344,#7330);
#7346=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7330,#7338);
#7347=CARTESIAN_POINT('', (205.0,-50.0,152.50000000000000));
#7348=DIRECTION('', (0.0,1.0,0.0));
#7349=AXIS1_PLACEMENT('', #7347,#7348);
#7350=SHAPE_ASPECT('', $,#38,.F.);
#7351=PROPERTY_DEFINITION('', $,#7350);
#7352=REPRESENTATION('', (#7349), #29);
#7353=PROPERTY_DEFINITION_REPRESENTATION(#7351,#7352);
#7354=DIMENSIONAL_LOCATION('', $,#143,#7350);
#7355=PRECISION_QUALIFIER(4);
#7361=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#735
5))REPRESENTATION_ITEM(''));
#7362=SHAPE_DIMENSION_REPRESENTATION('', (#7361), #29);
#7363=PRECISION_QUALIFIER(4);
#7364=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#7365=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#7366=MEASURE_QUALIFICATION('', '#7364', (#7363));
#7367=MEASURE_QUALIFICATION('', '#7365', (#7363));

```

```

#7368=TOLERANCE_VALUE(#7365,#7364);
#7369=PLUS_MINUS_TOLERANCE(#7368,#7354);
#7370=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7354,#7362);
#7377=(CHARACTERIZED_OBJECT('compound feature in
solid','counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid','counterbore',#38,.T.));
#7378=PROPERTY_DEFINITION('',$,#7377);
#7379=PRODUCT_DEFINITION_SHAPE('',$,#7377);
#7380=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#7381=DIRECTION('',(0.0,-1.0,0.0));
#7382=DIRECTION('',(-1.0,0.0,0.0));
#7383=AXIS2_PLACEMENT_3D('orientation',#7380,#7381,#7382);
#7384=REPRESENTATION('',(#2988,#3028,#3052,#3068,#3084),#29);
#7385=PROPERTY_DEFINITION_REPRESENTATION(#7378,#7384);
#7386=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7383),#29);
#7387=PROPERTY_DEFINITION_REPRESENTATION(#7378,#7386);
#7393=(CHARACTERIZED_OBJECT('',)$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#7394=PROPERTY_DEFINITION('',$,#7393);
#7395=PRODUCT_DEFINITION_SHAPE('',$,#7393);
#7396=CARTESIAN_POINT('',(25.0,0.0,27.500000000000000));
#7397=DIRECTION('',(0.0,-1.0,0.0));
#7398=DIRECTION('',(-1.0,0.0,0.0));
#7399=AXIS2_PLACEMENT_3D('orientation',#7396,#7397,#7398);
#7400=REPRESENTATION('',(#2988,#3068),#29);
#7401=PROPERTY_DEFINITION_REPRESENTATION(#7394,#7400);
#7402=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399),#29);
#7403=PROPERTY_DEFINITION_REPRESENTATION(#7394,#7402);
#7404=FEATURE_COMPONENT_DEFINITION('',)$);
#7405=PRODUCT_DEFINITION_SHAPE('',$,#7404);
#7406=CIRCULAR_CLOSED_PROFILE('circular profile',$,#7405,.F.);
#7407=STANDARD_UNCERTAINTY('upper limit','plus',0.250000000000000);
#7408=STANDARD_UNCERTAINTY('lower limit','minus',-0.250000000000000);
#7409=PRECISION_QUALIFIER(4);
#7415=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7407,#7408,#7409)
)REPRESENTATION_ITEM('diameter'));
#7416=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399,#7415),#29);
#7417=PROPERTY_DEFINITION('',$,#7406);
#7418=PROPERTY_DEFINITION_REPRESENTATION(#7417,#7416);
#7419=FEATURE_COMPONENT_DEFINITION('',)$);
#7420=PRODUCT_DEFINITION_SHAPE('',$,#7419);
#7421=PATH_FEATURE_COMPONENT('linear path','linear',#7420,.F.);
#7422=STANDARD_UNCERTAINTY('upper limit','plus',0.100000000000000);
#7423=STANDARD_UNCERTAINTY('lower limit','minus',-0.100000000000000);
#7424=PRECISION_QUALIFIER(4);
#7430=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7422,#7423,#7424)
)REPRESENTATION_ITEM('distance'));
#7431=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#7399,#7430),#29);
#7432=PROPERTY_DEFINITION('',$,#7421);
#7433=PROPERTY_DEFINITION_REPRESENTATION(#7432,#7431);

```

```

#7434=DIRECTION_SHAPE_REPRESENTATION('', (#7397), #29);
#7435=PROPERTY_DEFINITION_REPRESENTATION(#7432, #7434);
#7436=SHAPE_ASPECT('', 'diameter occurrence', #7395, .T.);
#7437=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7406, #7436);
#7438=SHAPE_ASPECT('', 'hole depth occurrence', #7395, .T.);
#7439=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #7421, #7438);

#7440=FEATURE_COMPONENT_DEFINITION('', $);
#7441=PRODUCT_DEFINITION_SHAPE('', $, #7440);
#7442=HOLE_BOTTOM('bottom condition', 'flat', #7441, .F.);
#7447=SHAPE_ASPECT('', 'bottom condition occurrence', #7395, .T.);

#7448=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7442, #7447);
#7449=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #7379, .T.);
#7455=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#7456=PROPERTY_DEFINITION('', $, #7455);
#7457=PRODUCT_DEFINITION_SHAPE('', $, #7455);
#7458=CARTESIAN_POINT('', (25.0, 0.0, 27.500000000000000));
#7459=DIRECTION('', (0.0, -1.0, 0.0));
#7460=DIRECTION('', (-1.0, 0.0, 0.0));
#7461=AXIS2_PLACEMENT_3D('orientation', #7458, #7459, #7460);
#7462=REPRESENTATION('', (#3028, #3052, #3084), #29);
#7463=PROPERTY_DEFINITION_REPRESENTATION(#7456, #7462);
#7464=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7461), #29);
#7465=PROPERTY_DEFINITION_REPRESENTATION(#7456, #7464);
#7466=FEATURE_COMPONENT_DEFINITION('', $);
#7467=PRODUCT_DEFINITION_SHAPE('', $, #7466);
#7468=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7467, .F.);
#7469=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.500000000000000);
#7470=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.500000000000000);
#7471=PRECISION_QUALIFIER(4);
#7477=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7469, #7470, #7471)
) REPRESENTATION_ITEM('diameter'));
#7478=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7461, #7477), #29);
#7479=PROPERTY_DEFINITION('', $, #7468);
#7480=PROPERTY_DEFINITION_REPRESENTATION(#7479, #7478);
#7481=FEATURE_COMPONENT_DEFINITION('', $);
#7482=PRODUCT_DEFINITION_SHAPE('', $, #7481);
#7483=PATH_FEATURE_COMPONENT('linear path', 'linear', #7482, .F.);
#7484=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#7485=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#7486=PRECISION_QUALIFIER(4);
#7492=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7484, #7485, #7486)
) REPRESENTATION_ITEM('distance'));
#7493=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7461, #7492), #29);
#7494=PROPERTY_DEFINITION('', $, #7483);
#7495=PROPERTY_DEFINITION_REPRESENTATION(#7494, #7493);
#7496=DIRECTION_SHAPE_REPRESENTATION('', (#7459), #29);

```

ISO 10303-223:2008 (E)

```

#7497=PROPERTY_DEFINITION_REPRESENTATION(#7494,#7496);
#7498=SHAPE_ASPECT('', 'diameter occurrence', #7457, .T.);
#7499=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7468, #7498);
#7500=SHAPE_ASPECT('', 'hole depth occurrence', #7457, .T.);
#7501=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #7483, #7500);

#7502=FEATURE_COMPONENT_DEFINITION('', $);
#7503=PRODUCT_DEFINITION_SHAPE('', $, #7502);
#7504=HOLE_BOTTOM('bottom condition', 'flat', #7503, .F.);
#7509=SHAPE_ASPECT('', 'bottom condition occurrence', #7457, .T.);

#7510=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7504, #7509);
#7511=FEATURE_COMPONENT_RELATIONSHIP('large hole', $, #7449, #7455);
#7512=FEATURE_COMPONENT_RELATIONSHIP('small hole', $, #7449, #7393);
#7513=CARTESIAN_POINT('', (25.0, -50.0, 27.500000000000000));
#7514=DIRECTION('', (0.0, 1.0, 0.0));
#7515=AXIS1_PLACEMENT('', #7513, #7514);
#7516=SHAPE_ASPECT('', $, #38, .F.);
#7517=PROPERTY_DEFINITION('', $, #7516);
#7518=REPRESENTATION('', (#7515), #29);
#7519=PROPERTY_DEFINITION_REPRESENTATION(#7517, #7518);
#7520=DIMENSIONAL_LOCATION('', $, #239, #7516);
#7521=PRECISION_QUALIFIER(4);
#7527=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7521))REPRESENTAT
ION_ITEM(''));
#7528=SHAPE_DIMENSION_REPRESENTATION('', (#7527), #29);
#7529=PRECISION_QUALIFIER(4);
#7530=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000), #10);
#7531=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000), #10);
#7532=MEASURE_QUALIFICATION('', '#7530', (#7529));
#7533=MEASURE_QUALIFICATION('', '#7531', (#7529));
#7534=TOLERANCE_VALUE(#7531, #7530);
#7535=PLUS_MINUS_TOLERANCE(#7534, #7520);
#7536=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7520, #7528);
#7537=CARTESIAN_POINT('', (25.0, -50.0, 27.500000000000000));
#7538=DIRECTION('', (0.0, 1.0, 0.0));
#7539=AXIS1_PLACEMENT('', #7537, #7538);
#7540=SHAPE_ASPECT('', $, #38, .F.);
#7541=PROPERTY_DEFINITION('', $, #7540);
#7542=REPRESENTATION('', (#7539), #29);
#7543=PROPERTY_DEFINITION_REPRESENTATION(#7541, #7542);
#7544=DIMENSIONAL_LOCATION('', $, #75, #7540);
#7545=PRECISION_QUALIFIER(4);
#7551=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.500000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#754
5))REPRESENTATION_ITEM(''));
#7552=SHAPE_DIMENSION_REPRESENTATION('', (#7551), #29);
#7553=PRECISION_QUALIFIER(4);
#7554=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000), #10);
#7555=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000), #10);

```



```

#7556=MEASURE_QUALIFICATION('', '#7554, (#7553));
#7557=MEASURE_QUALIFICATION('', '#7555, (#7553));
#7558=TOLERANCE_VALUE(#7555, #7554);
#7559=PLUS_MINUS_TOLERANCE(#7558, #7544);
#7560=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7544, #7552);
#7567=(CHARACTERIZED_OBJECT('compound feature in
solid', 'counterbore')COMPOSITE_HOLE()COMPOUND_FEATURE()FEATURE_DEFINITION()IN
STANCED_FEATURE()SHAPE_ASPECT('compound feature in
solid', 'counterbore', #38, .T.));
#7568=PROPERTY_DEFINITION('', $, #7567);
#7569=PRODUCT_DEFINITION_SHAPE('', $, #7567);
#7570=CARTESIAN_POINT('', (25.0, 0.0, 152.50000000000000));
#7571=DIRECTION('', (0.0, -1.0, 0.0));
#7572=DIRECTION('', (-1.0, 0.0, 0.0));
#7573=AXIS2_PLACEMENT_3D('orientation', #7570, #7571, #7572);
#7574=REPRESENTATION('', (#3118, #3158, #3182, #3198, #3214), #29);
#7575=PROPERTY_DEFINITION_REPRESENTATION(#7568, #7574);
#7576=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7573), #29);
#7577=PROPERTY_DEFINITION_REPRESENTATION(#7568, #7576);
#7583=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('', $, #38, .T.));
#7584=PROPERTY_DEFINITION('', $, #7583);
#7585=PRODUCT_DEFINITION_SHAPE('', $, #7583);
#7586=CARTESIAN_POINT('', (25.0, 0.0, 152.50000000000000));
#7587=DIRECTION('', (0.0, -1.0, 0.0));
#7588=DIRECTION('', (-1.0, 0.0, 0.0));
#7589=AXIS2_PLACEMENT_3D('orientation', #7586, #7587, #7588);
#7590=REPRESENTATION('', (#3118, #3198), #29);
#7591=PROPERTY_DEFINITION_REPRESENTATION(#7584, #7590);
#7592=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7589), #29);
#7593=PROPERTY_DEFINITION_REPRESENTATION(#7584, #7592);
#7594=FEATURE_COMPONENT_DEFINITION('', $);
#7595=PRODUCT_DEFINITION_SHAPE('', $, #7594);
#7596=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7595, .F.);
#7597=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2500000000000000);
#7598=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.2500000000000000);
#7599=PRECISION_QUALIFIER(4);
#7605=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(18.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7597, #7598, #7599)
)REPRESENTATION_ITEM('diameter'));
#7606=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7589, #7605), #29);
#7607=PROPERTY_DEFINITION('', $, #7596);
#7608=PROPERTY_DEFINITION_REPRESENTATION(#7607, #7606);
#7609=FEATURE_COMPONENT_DEFINITION('', $);
#7610=PRODUCT_DEFINITION_SHAPE('', $, #7609);
#7611=PATH_FEATURE_COMPONENT('linear path', 'linear', #7610, .F.);
#7612=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#7613=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#7614=PRECISION_QUALIFIER(4);
#7620=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7612, #7613, #7614)
)REPRESENTATION_ITEM('distance'));
#7621=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7589, #7620), #29);

```

```

#7622=PROPERTY_DEFINITION('', $, #7611);
#7623=PROPERTY_DEFINITION_REPRESENTATION(#7622, #7621);
#7624=DIRECTION_SHAPE_REPRESENTATION('', (#7587), #29);
#7625=PROPERTY_DEFINITION_REPRESENTATION(#7622, #7624);
#7626=SHAPE_ASPECT('', 'diameter occurrence', #7585, .T.);
#7627=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7596, #7626);
#7628=SHAPE_ASPECT('', 'hole depth occurrence', #7585, .T.);
#7629=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #7611, #7628);

#7630=FEATURE_COMPONENT_DEFINITION('', $);
#7631=PRODUCT_DEFINITION_SHAPE('', $, #7630);
#7632=HOLE_BOTTOM('bottom condition', 'flat', #7631, .F.);
#7637=SHAPE_ASPECT('', 'bottom condition occurrence', #7585, .T.);

#7638=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7632, #7637);
#7639=COMPOSITE_SHAPE_ASPECT('compound feature in solid', $, #7569, .T.);
#7645=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND
_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#7646=PROPERTY_DEFINITION('', $, #7645);
#7647=PRODUCT_DEFINITION_SHAPE('', $, #7645);
#7648=CARTESIAN_POINT('', (25.0, 0.0, 152.50000000000000));
#7649=DIRECTION('', (0.0, -1.0, 0.0));
#7650=DIRECTION('', (-1.0, 0.0, 0.0));
#7651=AXIS2_PLACEMENT_3D('orientation', #7648, #7649, #7650);
#7652=REPRESENTATION('', (#3158, #3182, #3214), #29);
#7653=PROPERTY_DEFINITION_REPRESENTATION(#7646, #7652);
#7654=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7651), #29);
#7655=PROPERTY_DEFINITION_REPRESENTATION(#7646, #7654);
#7656=FEATURE_COMPONENT_DEFINITION('', $);
#7657=PRODUCT_DEFINITION_SHAPE('', $, #7656);
#7658=CIRCULAR_CLOSED_PROFILE('circular profile', $, #7657, .F.);
#7659=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.500000000000000);
#7660=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.500000000000000);
#7661=PRECISION_QUALIFIER(4);
#7667=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(26.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7659, #7660, #7661)
) REPRESENTATION_ITEM('diameter'));
#7668=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7651, #7667), #29);
#7669=PROPERTY_DEFINITION('', $, #7658);
#7670=PROPERTY_DEFINITION_REPRESENTATION(#7669, #7668);
#7671=FEATURE_COMPONENT_DEFINITION('', $);
#7672=PRODUCT_DEFINITION_SHAPE('', $, #7671);
#7673=PATH_FEATURE_COMPONENT('linear path', 'linear', #7672, .F.);
#7674=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#7675=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#7676=PRECISION_QUALIFIER(4);
#7682=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7674, #7675, #7676)
) REPRESENTATION_ITEM('distance'));
#7683=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7651, #7682), #29);
#7684=PROPERTY_DEFINITION('', $, #7673);

```

```

#7685=PROPERTY_DEFINITION_REPRESENTATION(#7684,#7683);
#7686=DIRECTION_SHAPE_REPRESENTATION('',(#7649),#29);
#7687=PROPERTY_DEFINITION_REPRESENTATION(#7684,#7686);
#7688=SHAPE_ASPECT('', 'diameter occurrence', #7647, .T.);
#7689=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #7658, #7688);
#7690=SHAPE_ASPECT('', 'hole depth occurrence', #7647, .T.);
#7691=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #7673, #7690);

#7692=FEATURE_COMPONENT_DEFINITION('', $);
#7693=PRODUCT_DEFINITION_SHAPE('', $, #7692);
#7694=HOLE_BOTTOM('bottom condition', 'flat', #7693, .F.);
#7699=SHAPE_ASPECT('', 'bottom condition occurrence', #7647, .T.);

#7700=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #7694, #7699);
#7701=FEATURE_COMPONENT_RELATIONSHIP('large hole', $, #7639, #7645);
#7702=FEATURE_COMPONENT_RELATIONSHIP('small hole', $, #7639, #7583);
#7703=CARTESIAN_POINT('', (25.0, -50.0, 152.50000000000000));
#7704=DIRECTION('', (0.0, 1.0, 0.0));
#7705=AXIS1_PLACEMENT('', #7703, #7704);
#7706=SHAPE_ASPECT('', $, #38, .F.);
#7707=PROPERTY_DEFINITION('', $, #7706);
#7708=REPRESENTATION('', (#7705), #29);
#7709=PROPERTY_DEFINITION_REPRESENTATION(#7707, #7708);
#7710=DIMENSIONAL_LOCATION('', $, #239, #7706);
#7711=PRECISION_QUALIFIER(4);
#7717=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7711))REPRESENTAT
ION_ITEM(''));
#7718=SHAPE_DIMENSION_REPRESENTATION('', (#7717), #29);
#7719=PRECISION_QUALIFIER(4);
#7720=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#7721=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#7722=MEASURE_QUALIFICATION('', '#7720, (#7719)');
#7723=MEASURE_QUALIFICATION('', '#7721, (#7719)');
#7724=TOLERANCE_VALUE(#7721, #7720);
#7725=PLUS_MINUS_TOLERANCE(#7724, #7710);
#7726=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7710, #7718);
#7727=CARTESIAN_POINT('', (25.0, -50.0, 152.50000000000000));
#7728=DIRECTION('', (0.0, 1.0, 0.0));
#7729=AXIS1_PLACEMENT('', #7727, #7728);
#7730=SHAPE_ASPECT('', $, #38, .F.);
#7731=PROPERTY_DEFINITION('', $, #7730);
#7732=REPRESENTATION('', (#7729), #29);
#7733=PROPERTY_DEFINITION_REPRESENTATION(#7731, #7732);
#7734=DIMENSIONAL_LOCATION('', $, #143, #7730);
#7735=PRECISION_QUALIFIER(4);
#7741=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(27.5000000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#773
5))REPRESENTATION_ITEM(''));
#7742=SHAPE_DIMENSION_REPRESENTATION('', (#7741), #29);
#7743=PRECISION_QUALIFIER(4);

```

```

#7744=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000),#10);
#7745=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0100000000000000),#10);
#7746=MEASURE_QUALIFICATION('', '#7744', (#7743));
#7747=MEASURE_QUALIFICATION('', '#7745', (#7743));
#7748=TOLERANCE_VALUE(#7745, #7744);
#7749=PLUS_MINUS_TOLERANCE(#7748, #7734);
#7750=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7734, #7742);
#7756=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() SHAPE
ASPECT('', $, #38, .T.) SLOT());
#7757=PROPERTY_DEFINITION('', $, #7756);
#7758=PRODUCT_DEFINITION_SHAPE('', $, #7756);
#7759=CARTESIAN_POINT('', (205.0, -10.0, 90.0));
#7760=DIRECTION('', (1.0, 0.0, 0.0));
#7761=DIRECTION('', (0.0, 0.0, -1.0));
#7762=AXIS2_PLACEMENT_3D('orientation', #7759, #7760, #7761);
#7763=REPRESENTATION('', (#3240, #3268, #3288, #3304), #29);
#7764=PROPERTY_DEFINITION_REPRESENTATION(#7757, #7763);
#7765=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7762), #29);
#7766=PROPERTY_DEFINITION_REPRESENTATION(#7757, #7765);
#7767=FEATURE_COMPONENT_DEFINITION('', $);
#7768=PRODUCT_DEFINITION_SHAPE('', $, #7767);
#7769=SQUARE_U_PROFILE('', $, #7768, .F.);
#7770=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#7771=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#7772=PRECISION_QUALIFIER(4);
#7778=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7770, #7771, #7772)
) REPRESENTATION_ITEM('width'));
#7783=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('first angle'));
#7788=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.
0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('second angle'));
#7789=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7762, #7778, #7783, #7788), #29);
#7790=PROPERTY_DEFINITION('', $, #7769);
#7791=PROPERTY_DEFINITION_REPRESENTATION(#7790, #7789);
#7792=SHAPE_ASPECT('', 'swept shape occurrence', #7758, .T.);
#7793=SHAPE_DEFINING_RELATIONSHIP('', 'profile usage', #7769, #7792);
#7794=FEATURE_COMPONENT_DEFINITION('', $);
#7795=PRODUCT_DEFINITION_SHAPE('', $, #7794);
#7796=PATH_FEATURE_COMPONENT('linear path', 'linear', #7795, .F.);
#7797=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#7798=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#7799=PRECISION_QUALIFIER(4);
#7805=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0), #10) QUALIFIED_REPRESENTATION_ITEM((#7797, #7798, #7799)
) REPRESENTATION_ITEM('distance'));
#7806=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7762, #7805), #29);
#7807=PROPERTY_DEFINITION('', $, #7796);
#7808=PROPERTY_DEFINITION_REPRESENTATION(#7807, #7806);
#7809=DIRECTION_SHAPE_REPRESENTATION('', (#7760), #29);
#7810=PROPERTY_DEFINITION_REPRESENTATION(#7807, #7809);
#7811=SHAPE_ASPECT('', 'course of travel occurrence', #7758, .T.);

```

```

#7812=SHAPE_DEFINING_RELATIONSHIP('course of travel','path feature component
usage',#7796,#7811);
#7813=FEATURE_COMPONENT_DEFINITION('', $);
#7814=PRODUCT_DEFINITION_SHAPE('', $,#7813);
#7815=SLOT_END('', 'radiused',#7814, .F.);
#7816=FEATURE_COMPONENT_DEFINITION('', $);
#7817=PRODUCT_DEFINITION_SHAPE('', $,#7816);
#7818=SLOT_END('', 'open',#7817, .F.);
#7819=SHAPE_ASPECT('', 'end condition occurrence',#7758, .T.);
#7820=FEATURE_COMPONENT_RELATIONSHIP('course of travel start','slot end
usage',#7815,#7819);
#7821=SHAPE_ASPECT('', 'end condition occurrence',#7758, .T.);
#7822=FEATURE_COMPONENT_RELATIONSHIP('course of travel end','slot end
usage',#7818,#7821);
#7823=DIMENSIONAL_LOCATION('', $,#1110,#3269);
#7824=PRECISION_QUALIFIER(4);
#7830=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7824))REPRESENTAT
ION_ITEM(''));
#7831=SHAPE_DIMENSION_REPRESENTATION('', (#7830),#29);
#7832=PRECISION_QUALIFIER(4);
#7833=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#7834=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#7835=MEASURE_QUALIFICATION('', '#7833', (#7832));
#7836=MEASURE_QUALIFICATION('', '#7834', (#7832));
#7837=TOLERANCE_VALUE(#7834,#7833);
#7838=PLUS_MINUS_TOLERANCE(#7837,#7823);
#7839=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7823,#7831);
#7840=DIMENSIONAL_LOCATION('', $,#5503,#3241);
#7841=PRECISION_QUALIFIER(4);
#7847=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7841))REPRESENTATI
ON_ITEM(''));
#7848=SHAPE_DIMENSION_REPRESENTATION('', (#7847),#29);
#7849=PRECISION_QUALIFIER(4);
#7850=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#7851=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#7852=MEASURE_QUALIFICATION('', '#7850', (#7849));
#7853=MEASURE_QUALIFICATION('', '#7851', (#7849));
#7854=TOLERANCE_VALUE(#7851,#7850);
#7855=PLUS_MINUS_TOLERANCE(#7854,#7840);
#7856=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7840,#7848);
#7862=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()SHAPE
_ASPECT('', $,#38, .T.)SLOT());
#7863=PROPERTY_DEFINITION('', $,#7862);
#7864=PRODUCT_DEFINITION_SHAPE('', $,#7862);
#7865=CARTESIAN_POINT('', (80.0,-10.0,90.0));
#7866=DIRECTION('', (1.0,0.0,0.0));
#7867=DIRECTION('', (0.0,0.0,-1.0));
#7868=AXIS2_PLACEMENT_3D('orientation',#7865,#7866,#7867);
#7869=REPRESENTATION('', (#3348,#3372,#3392,#3412,#3428),#29);
#7870=PROPERTY_DEFINITION_REPRESENTATION(#7863,#7869);
#7871=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7868),#29);

```

```

#7872=PROPERTY_DEFINITION_REPRESENTATION(#7863,#7871);
#7873=FEATURE_COMPONENT_DEFINITION('', $);
#7874=PRODUCT_DEFINITION_SHAPE('', $, #7873);
#7875=SQUARE_U_PROFILE('', $, #7874, .F.);
#7876=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.2000000000000000);
#7877=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0);
#7878=PRECISION_QUALIFIER(4);
#7884=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0), #10)QUALIFIED_REPRESENTATION_ITEM((#7876, #7877, #7878)REPRESENTATION_ITEM('width')));
#7889=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('first angle'));
#7894=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19)PLANE_ANGLE_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('second angle'));
#7895=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7868, #7884, #7889, #7894), #29);
#7896=PROPERTY_DEFINITION('', $, #7875);
#7897=PROPERTY_DEFINITION_REPRESENTATION(#7896, #7895);
#7898=SHAPE_ASPECT('', 'swept shape occurrence', #7864, .T.);
#7899=SHAPE_DEFINING_RELATIONSHIP('', 'profile usage', #7875, #7898);
#7900=FEATURE_COMPONENT_DEFINITION('', $);
#7901=PRODUCT_DEFINITION_SHAPE('', $, #7900);
#7902=PATH_FEATURE_COMPONENT('linear path', 'linear', #7901, .F.);
#7907=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(LENGTH_MEASURE(35.0), #10)REPRESENTATION_ITEM('distance'));
#7908=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7868, #7907), #29);
#7909=PROPERTY_DEFINITION('', $, #7902);
#7910=PROPERTY_DEFINITION_REPRESENTATION(#7909, #7908);
#7911=DIRECTION_SHAPE_REPRESENTATION('', (#7866), #29);
#7912=PROPERTY_DEFINITION_REPRESENTATION(#7909, #7911);
#7913=SHAPE_ASPECT('', 'course of travel occurrence', #7864, .T.);
#7914=SHAPE_DEFINING_RELATIONSHIP('course of travel', 'path feature component usage', #7902, #7913);
#7915=FEATURE_COMPONENT_DEFINITION('', $);
#7916=PRODUCT_DEFINITION_SHAPE('', $, #7915);
#7917=SLOT_END('', 'radiused', #7916, .F.);
#7918=FEATURE_COMPONENT_DEFINITION('', $);
#7919=PRODUCT_DEFINITION_SHAPE('', $, #7918);
#7920=SLOT_END('', 'radiused', #7919, .F.);
#7921=SHAPE_ASPECT('', 'end condition occurrence', #7864, .T.);
#7922=FEATURE_COMPONENT_RELATIONSHIP('course of travel start', 'slot end usage', #7917, #7921);
#7923=SHAPE_ASPECT('', 'end condition occurrence', #7864, .T.);
#7924=FEATURE_COMPONENT_RELATIONSHIP('course of travel end', 'slot end usage', #7920, #7923);
#7925=CARTESIAN_POINT('', (260.0, 0.0, 0.0));
#7926=DIRECTION('', (0.0, 1.0, 0.0));
#7927=DIRECTION('', (0.0, 0.0, 1.0));
#7928=AXIS2_PLACEMENT_3D('', #7925, #7926, #7927);
#7929=SHAPE_ASPECT('', $, #38, .F.);
#7930=PROPERTY_DEFINITION('', $, #7929);
#7931=REPRESENTATION('', (#7928), #29);
#7932=PROPERTY_DEFINITION_REPRESENTATION(#7930, #7931);
#7933=DIMENSIONAL_LOCATION('', $, #7929, #3349);

```

```

#7934=PRECISION_QUALIFIER(4);
#7940=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7934))REPRESENTAT
ION_ITEM(''));
#7941=SHAPE_DIMENSION_REPRESENTATION('',( #7940),#29);
#7942=PRECISION_QUALIFIER(4);
#7943=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#7944=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.2500000000000000),#10);
#7945=MEASURE_QUALIFICATION('','',#7943,(#7942));
#7946=MEASURE_QUALIFICATION('','',#7944,(#7942));
#7947=TOLERANCE_VALUE(#7944,#7943);
#7948=PLUS_MINUS_TOLERANCE(#7947,#7933);
#7949=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7933,#7941);
#7950=DIMENSIONAL_LOCATION('','$,#3429,#3393);
#7951=PRECISION_QUALIFIER(4);
#7957=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(25.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7951))REPRESENTAT
ION_ITEM(''));
#7958=SHAPE_DIMENSION_REPRESENTATION('',( #7957),#29);
#7959=PRECISION_QUALIFIER(4);
#7960=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.4000000000000000),#10);
#7961=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0),#10);
#7962=MEASURE_QUALIFICATION('','',#7960,(#7959));
#7963=MEASURE_QUALIFICATION('','',#7961,(#7959));
#7964=TOLERANCE_VALUE(#7961,#7960);
#7965=PLUS_MINUS_TOLERANCE(#7964,#7950);
#7966=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7950,#7958);
#7967=DIMENSIONAL_LOCATION('','$,#1261,#3429);
#7968=PRECISION_QUALIFIER(4);
#7974=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0),#10)QUALIFIED_REPRESENTATION_ITEM((#7968))REPRESENTA
TION_ITEM(''));
#7975=SHAPE_DIMENSION_REPRESENTATION('',( #7974),#29);
#7976=PRECISION_QUALIFIER(4);
#7977=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#7978=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#7979=MEASURE_QUALIFICATION('','',#7977,(#7976));
#7980=MEASURE_QUALIFICATION('','',#7978,(#7976));
#7981=TOLERANCE_VALUE(#7978,#7977);
#7982=PLUS_MINUS_TOLERANCE(#7981,#7967);
#7983=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#7967,#7975);
#7989=(CHARACTERIZED_OBJECT('','$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('','$,#38,.T.));
#7990=PROPERTY_DEFINITION('','$,#7989);
#7991=PRODUCT_DEFINITION_SHAPE('','$,#7989);
#7992=CARTESIAN_POINT('',(190.0,0.0,120.0));
#7993=DIRECTION('',(0.0,-1.0,0.0));
#7994=DIRECTION('',(1.0,0.0,0.0));
#7995=AXIS2_PLACEMENT_3D('orientation',#7992,#7993,#7994);
#7996=REPRESENTATION('',( #3458,#3482,#3498),#29);
#7997=PROPERTY_DEFINITION_REPRESENTATION(#7990,#7996);
#7998=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #7995),#29);
#7999=PROPERTY_DEFINITION_REPRESENTATION(#7990,#7998);

```

ISO 10303-223:2008 (E)

```

#8000=FEATURE_COMPONENT_DEFINITION('', $);
#8001=PRODUCT_DEFINITION_SHAPE('', $, #8000);
#8002=CIRCULAR_CLOSED_PROFILE('circular profile', $, #8001, .F.);
#8003=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1201000000000000);
#8004=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1201000000000000);
#8005=PRECISION_QUALIFIER(4);
#8011=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.1377000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#8003
, #8004, #8005))REPRESENTATION_ITEM('diameter'));
#8012=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7995, #8011), #29);
#8013=PROPERTY_DEFINITION('', $, #8002);
#8014=PROPERTY_DEFINITION_REPRESENTATION(#8013, #8012);
#8015=FEATURE_COMPONENT_DEFINITION('', $);
#8016=PRODUCT_DEFINITION_SHAPE('', $, #8015);
#8017=PATH_FEATURE_COMPONENT('linear path', 'linear', #8016, .F.);
#8018=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#8019=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#8020=PRECISION_QUALIFIER(4);
#8026=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001), #10)QUALIFIED_REPRESENTATION_ITEM((#801
8, #8019, #8020))REPRESENTATION_ITEM('distance'));
#8027=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#7995, #8026), #29);
#8028=PROPERTY_DEFINITION('', $, #8017);
#8029=PROPERTY_DEFINITION_REPRESENTATION(#8028, #8027);
#8030=DIRECTION_SHAPE_REPRESENTATION('', (#7993), #29);
#8031=PROPERTY_DEFINITION_REPRESENTATION(#8028, #8030);
#8032=SHAPE_ASPECT('', 'diameter occurrence', #7991, .T.);
#8033=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #8002, #8032);
#8034=SHAPE_ASPECT('', 'hole depth occurrence', #7991, .T.);
#8035=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #8017, #8034);

#8036=FEATURE_COMPONENT_DEFINITION('', $);
#8037=PRODUCT_DEFINITION_SHAPE('', $, #8036);
#8038=HOLE_BOTTOM('bottom condition', 'flat', #8037, .F.);
#8043=SHAPE_ASPECT('', 'bottom condition occurrence', #7991, .T.);

#8044=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #8038, #8043);
#8045=CARTESIAN_POINT('', (190.0, 0.0, 120.0));
#8046=DIRECTION('', (0.0, -1.0, 0.0));
#8047=AXIS1_PLACEMENT('', #8045, #8046);
#8048=SHAPE_ASPECT('', $, #38, .F.);
#8049=PROPERTY_DEFINITION('', $, #8048);
#8050=REPRESENTATION('', (#8047), #29);
#8051=PROPERTY_DEFINITION_REPRESENTATION(#8049, #8050);
#8052=DIMENSIONAL_LOCATION('', $, #1261, #8048);
#8053=PRECISION_QUALIFIER(4);
#8059=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8053))REPRESENTAT
ION_ITEM(''));
#8060=SHAPE_DIMENSION_REPRESENTATION('', (#8059), #29);
#8061=PRECISION_QUALIFIER(4);

```



```

#8062=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8063=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8064=MEASURE_QUALIFICATION('','',#8062,(#8061));
#8065=MEASURE_QUALIFICATION('','',#8063,(#8061));
#8066=TOLERANCE_VALUE(#8063,#8062);
#8067=PLUS_MINUS_TOLERANCE(#8066,#8052);
#8068=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8052,#8060);
#8069=CARTESIAN_POINT('',(190.0,0.0,120.0));
#8070=DIRECTION('',(0.0,-1.0,0.0));
#8071=AXIS1_PLACEMENT('',#8069,#8070);
#8072=SHAPE_ASPECT('',$,#38,.F.);
#8073=PROPERTY_DEFINITION('',$,#8072);
#8074=REPRESENTATION('',(8071),#29);
#8075=PROPERTY_DEFINITION_REPRESENTATION(#8073,#8074);
#8076=DIMENSIONAL_LOCATION('',$,#143,#8072);
#8077=PRECISION_QUALIFIER(4);
#8083=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(60.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8077))REPRESENTAT
ION_ITEM(''));
#8084=SHAPE_DIMENSION_REPRESENTATION('',(8083),#29);
#8085=PRECISION_QUALIFIER(4);
#8086=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8087=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8088=MEASURE_QUALIFICATION('','',#8086,(#8085));
#8089=MEASURE_QUALIFICATION('','',#8087,(#8085));
#8090=TOLERANCE_VALUE(#8087,#8086);
#8091=PLUS_MINUS_TOLERANCE(#8090,#8076);
#8092=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8076,#8084);
#8098=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8099=PROPERTY_DEFINITION('',$,#8098);
#8100=PRODUCT_DEFINITION_SHAPE('',$,#8098);
#8101=CARTESIAN_POINT('',(190.0,0.0,60.0));
#8102=DIRECTION('',(0.0,-1.0,0.0));
#8103=DIRECTION('',(1.0,0.0,0.0));
#8104=AXIS2_PLACEMENT_3D('orientation',#8101,#8102,#8103);
#8105=REPRESENTATION('',(3528,#3552,#3568),#29);
#8106=PROPERTY_DEFINITION_REPRESENTATION(#8099,#8105);
#8107=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(8104),#29);
#8108=PROPERTY_DEFINITION_REPRESENTATION(#8099,#8107);
#8109=FEATURE_COMPONENT_DEFINITION('',$);
#8110=PRODUCT_DEFINITION_SHAPE('',$,#8109);
#8111=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8110,.F.);
#8112=STANDARD_UNCERTAINTY('upper limit','plus',0.1201000000000000);
#8113=STANDARD_UNCERTAINTY('lower limit','minus',-0.1201000000000000);
#8114=PRECISION_QUALIFIER(4);
#8120=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.1377000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8112
,#8113,#8114))REPRESENTATION_ITEM('diameter'));
#8121=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(8104,#8120),#29);
#8122=PROPERTY_DEFINITION('',$,#8111);
#8123=PROPERTY_DEFINITION_REPRESENTATION(#8122,#8121);
#8124=FEATURE_COMPONENT_DEFINITION('',$);

```

## ISO 10303-223:2008 (E)

```
#8125=PRODUCT_DEFINITION_SHAPE('', $, #8124);
#8126=PATH_FEATURE_COMPONENT('linear path', 'linear', #8125, .F.);
#8127=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#8128=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#8129=PRECISION_QUALIFIER(4);
#8135=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.050000000000001), #10)QUALIFIED_REPRESENTATION_ITEM((#812
7, #8128, #8129))REPRESENTATION_ITEM('distance'));
#8136=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8104, #8135), #29);
#8137=PROPERTY_DEFINITION('', $, #8126);
#8138=PROPERTY_DEFINITION_REPRESENTATION(#8137, #8136);
#8139=DIRECTION_SHAPE_REPRESENTATION('', (#8102), #29);
#8140=PROPERTY_DEFINITION_REPRESENTATION(#8137, #8139);
#8141=SHAPE_ASPECT('', 'diameter occurrence', #8100, .T.);
#8142=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #8111, #8141);
#8143=SHAPE_ASPECT('', 'hole depth occurrence', #8100, .T.);
#8144=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #8126, #8143);

#8145=FEATURE_COMPONENT_DEFINITION('', $);
#8146=PRODUCT_DEFINITION_SHAPE('', $, #8145);
#8147=HOLE_BOTTOM('bottom condition', 'flat', #8146, .F.);
#8152=SHAPE_ASPECT('', 'bottom condition occurrence', #8100, .T.);

#8153=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #8147, #8152);
#8154=CARTESIAN_POINT('', (190.0, 0.0, 60.0));
#8155=DIRECTION('', (0.0, -1.0, 0.0));
#8156=AXIS1_PLACEMENT('', #8154, #8155);
#8157=SHAPE_ASPECT('', $, #38, .F.);
#8158=PROPERTY_DEFINITION('', $, #8157);
#8159=REPRESENTATION('', (#8156), #29);
#8160=PROPERTY_DEFINITION_REPRESENTATION(#8158, #8159);
#8161=DIMENSIONAL_LOCATION('', $, #1261, #8157);
#8162=PRECISION_QUALIFIER(4);
#8168=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(40.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8162))REPRESENTAT
ION_ITEM(''));
#8169=SHAPE_DIMENSION_REPRESENTATION('', (#8168), #29);
#8170=PRECISION_QUALIFIER(4);
#8171=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8172=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8173=MEASURE_QUALIFICATION('', '#8171', (#8170));
#8174=MEASURE_QUALIFICATION('', '#8172', (#8170));
#8175=TOLERANCE_VALUE(#8172, #8171);
#8176=PLUS_MINUS_TOLERANCE(#8175, #8161);
#8177=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8161, #8169);
#8178=CARTESIAN_POINT('', (190.0, 0.0, 60.0));
#8179=DIRECTION('', (0.0, -1.0, 0.0));
#8180=AXIS1_PLACEMENT('', #8178, #8179);
#8181=SHAPE_ASPECT('', $, #38, .F.);
#8182=PROPERTY_DEFINITION('', $, #8181);
#8183=REPRESENTATION('', (#8180), #29);
```

```

#8184=PROPERTY_DEFINITION_REPRESENTATION(#8182,#8183);
#8185=DIMENSIONAL_LOCATION('', $, #143, #8181);
#8186=PRECISION_QUALIFIER(4);
#8192=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8186))REPRESENTA
TION_ITEM(''));
#8193=SHAPE_DIMENSION_REPRESENTATION('', (#8192), #29);
#8194=PRECISION_QUALIFIER(4);
#8195=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8196=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8197=MEASURE_QUALIFICATION('', '#8195, (#8194));
#8198=MEASURE_QUALIFICATION('', '#8196, (#8194));
#8199=TOLERANCE_VALUE(#8196, #8195);
#8200=PLUS_MINUS_TOLERANCE(#8199, #8185);
#8201=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8185, #8193);
#8207=(CHARACTERIZED_OBJECT('', $)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('', $, #38, .T.));
#8208=PROPERTY_DEFINITION('', $, #8207);
#8209=PRODUCT_DEFINITION_SHAPE('', $, #8207);
#8210=CARTESIAN_POINT('', (130.0, 0.0, 120.0));
#8211=DIRECTION('', (0.0, -1.0, 0.0));
#8212=DIRECTION('', (1.0, 0.0, 0.0));
#8213=AXIS2_PLACEMENT_3D('orientation', #8210, #8211, #8212);
#8214=REPRESENTATION('', (#3598, #3622, #3638), #29);
#8215=PROPERTY_DEFINITION_REPRESENTATION(#8208, #8214);
#8216=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8213), #29);
#8217=PROPERTY_DEFINITION_REPRESENTATION(#8208, #8216);
#8218=FEATURE_COMPONENT_DEFINITION('', $);
#8219=PRODUCT_DEFINITION_SHAPE('', $, #8218);
#8220=CIRCULAR_CLOSED_PROFILE('circular profile', $, #8219, .F.);
#8221=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1201000000000000);
#8222=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1201000000000000);
#8223=PRECISION_QUALIFIER(4);
#8229=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.1377000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#8221
, #8222, #8223))REPRESENTATION_ITEM('diameter'));
#8230=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8213, #8229), #29);
#8231=PROPERTY_DEFINITION('', $, #8220);
#8232=PROPERTY_DEFINITION_REPRESENTATION(#8231, #8230);
#8233=FEATURE_COMPONENT_DEFINITION('', $);
#8234=PRODUCT_DEFINITION_SHAPE('', $, #8233);
#8235=PATH_FEATURE_COMPONENT('linear path', 'linear', #8234, .F.);
#8236=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#8237=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#8238=PRECISION_QUALIFIER(4);
#8244=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001), #10)QUALIFIED_REPRESENTATION_ITEM((#823
6, #8237, #8238))REPRESENTATION_ITEM('distance'));
#8245=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8213, #8244), #29);
#8246=PROPERTY_DEFINITION('', $, #8235);
#8247=PROPERTY_DEFINITION_REPRESENTATION(#8246, #8245);
#8248=DIRECTION_SHAPE_REPRESENTATION('', (#8211), #29);
#8249=PROPERTY_DEFINITION_REPRESENTATION(#8246, #8248);

```

## ISO 10303-223:2008 (E)

```
#8250=SHAPE_ASPECT('', 'diameter occurrence', #8209, .T.);
#8251=SHAPE_DEFINING_RELATIONSHIP('diameter', 'profile usage', #8220, #8250);
#8252=SHAPE_ASPECT('', 'hole depth occurrence', #8209, .T.);
#8253=SHAPE_DEFINING_RELATIONSHIP('hole depth', 'path feature component
usage', #8235, #8252);

#8254=FEATURE_COMPONENT_DEFINITION('', $);
#8255=PRODUCT_DEFINITION_SHAPE('', $, #8254);
#8256=HOLE_BOTTOM('bottom condition', 'flat', #8255, .F.);
#8261=SHAPE_ASPECT('', 'bottom condition occurrence', #8209, .T.);

#8262=FEATURE_COMPONENT_RELATIONSHIP('hole depth end', 'hole bottom
usage', #8256, #8261);
#8263=CARTESIAN_POINT('', (130.0, 0.0, 120.0));
#8264=DIRECTION('', (0.0, -1.0, 0.0));
#8265=AXIS1_PLACEMENT('', #8263, #8264);
#8266=SHAPE_ASPECT('', $, #38, .F.);
#8267=PROPERTY_DEFINITION('', $, #8266);
#8268=REPRESENTATION('', (#8265), #29);
#8269=PROPERTY_DEFINITION_REPRESENTATION(#8267, #8268);
#8270=DIMENSIONAL_LOCATION('', $, #1261, #8266);
#8271=PRECISION_QUALIFIER(4);
#8277=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(100.0), #10) QUALIFIED_REPRESENTATION_ITEM((#8271)) REPRESENTA
TION_ITEM(''));
#8278=SHAPE_DIMENSION_REPRESENTATION('', (#8277), #29);
#8279=PRECISION_QUALIFIER(4);
#8280=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8281=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8282=MEASURE_QUALIFICATION('', '#8280, (#8279));
#8283=MEASURE_QUALIFICATION('', '#8281, (#8279));
#8284=TOLERANCE_VALUE(#8281, #8280);
#8285=PLUS_MINUS_TOLERANCE(#8284, #8270);
#8286=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8270, #8278);
#8287=CARTESIAN_POINT('', (130.0, 0.0, 120.0));
#8288=DIRECTION('', (0.0, -1.0, 0.0));
#8289=AXIS1_PLACEMENT('', #8287, #8288);
#8290=SHAPE_ASPECT('', $, #38, .F.);
#8291=PROPERTY_DEFINITION('', $, #8290);
#8292=REPRESENTATION('', (#8289), #29);
#8293=PROPERTY_DEFINITION_REPRESENTATION(#8291, #8292);
#8294=DIMENSIONAL_LOCATION('', $, #143, #8290);
#8295=PRECISION_QUALIFIER(4);
#8301=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(60.0), #10) QUALIFIED_REPRESENTATION_ITEM((#8295)) REPRESENTAT
ION_ITEM(''));
#8302=SHAPE_DIMENSION_REPRESENTATION('', (#8301), #29);
#8303=PRECISION_QUALIFIER(4);
#8304=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8305=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8306=MEASURE_QUALIFICATION('', '#8304, (#8303));
#8307=MEASURE_QUALIFICATION('', '#8305, (#8303));
#8308=TOLERANCE_VALUE(#8305, #8304);
```

```

#8309=PLUS_MINUS_TOLERANCE(#8308,#8294);
#8310=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8294,#8302);
#8316=(CHARACTERIZED_OBJECT('',$)FEATURE_DEFINITION()INSTANCED_FEATURE()ROUND
_HOLE()SHAPE_ASPECT('',$,#38,.T.));
#8317=PROPERTY_DEFINITION('',$,#8316);
#8318=PRODUCT_DEFINITION_SHAPE('',$,#8316);
#8319=CARTESIAN_POINT('',(130.0,0.0,60.0));
#8320=DIRECTION('',(0.0,-1.0,0.0));
#8321=DIRECTION('',(1.0,0.0,0.0));
#8322=AXIS2_PLACEMENT_3D('orientation',#8319,#8320,#8321);
#8323=REPRESENTATION('',(3668,#3692,#3708),#29);
#8324=PROPERTY_DEFINITION_REPRESENTATION(#8317,#8323);
#8325=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(8322),#29);
#8326=PROPERTY_DEFINITION_REPRESENTATION(#8317,#8325);
#8327=FEATURE_COMPONENT_DEFINITION('',$);
#8328=PRODUCT_DEFINITION_SHAPE('',$,#8327);
#8329=CIRCULAR_CLOSED_PROFILE('circular profile',$,#8328,.F.);
#8330=STANDARD_UNCERTAINTY('upper limit','plus',0.1201000000000000);
#8331=STANDARD_UNCERTAINTY('lower limit','minus',-0.1201000000000000);
#8332=PRECISION_QUALIFIER(4);
#8338=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.1377000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#8330
,#8331,#8332))REPRESENTATION_ITEM('diameter'));
#8339=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(8322,#8338),#29);
#8340=PROPERTY_DEFINITION('',$,#8329);
#8341=PROPERTY_DEFINITION_REPRESENTATION(#8340,#8339);
#8342=FEATURE_COMPONENT_DEFINITION('',$);
#8343=PRODUCT_DEFINITION_SHAPE('',$,#8342);
#8344=PATH_FEATURE_COMPONENT('linear path','linear',#8343,.F.);
#8345=STANDARD_UNCERTAINTY('upper limit','plus',0.1000000000000000);
#8346=STANDARD_UNCERTAINTY('lower limit','minus',-0.1000000000000000);
#8347=PRECISION_QUALIFIER(4);
#8353=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.0500000000000001),#10)QUALIFIED_REPRESENTATION_ITEM((#834
5,#8346,#8347))REPRESENTATION_ITEM('distance'));
#8354=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(8322,#8353),#29);
#8355=PROPERTY_DEFINITION('',$,#8344);
#8356=PROPERTY_DEFINITION_REPRESENTATION(#8355,#8354);
#8357=DIRECTION_SHAPE_REPRESENTATION('',(8320),#29);
#8358=PROPERTY_DEFINITION_REPRESENTATION(#8355,#8357);
#8359=SHAPE_ASPECT('','diameter occurrence',#8318,.T.);
#8360=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8329,#8359);
#8361=SHAPE_ASPECT('','hole depth occurrence',#8318,.T.);
#8362=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8344,#8361);

#8363=FEATURE_COMPONENT_DEFINITION('',$);
#8364=PRODUCT_DEFINITION_SHAPE('',$,#8363);
#8365=HOLE_BOTTOM('bottom condition','flat',#8364,.F.);
#8370=SHAPE_ASPECT('','bottom condition occurrence',#8318,.T.);

#8371=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8365,#8370);

```

ISO 10303-223:2008 (E)

```

#8372=CARTESIAN_POINT('', (130.0, 0.0, 60.0));
#8373=DIRECTION('', (0.0, -1.0, 0.0));
#8374=AXIS1_PLACEMENT('', #8372, #8373);
#8375=SHAPE_ASPECT('', $, #38, .F.);
#8376=PROPERTY_DEFINITION('', $, #8375);
#8377=REPRESENTATION('', (#8374), #29);
#8378=PROPERTY_DEFINITION_REPRESENTATION(#8376, #8377);
#8379=DIMENSIONAL_LOCATION('', $, #1261, #8375);
#8380=PRECISION_QUALIFIER(4);
#8386=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(100.0), #10) QUALIFIED_REPRESENTATION_ITEM((#8380)) REPRESENTA
TION_ITEM(''));
#8387=SHAPE_DIMENSION_REPRESENTATION('', (#8386), #29);
#8388=PRECISION_QUALIFIER(4);
#8389=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8390=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8391=MEASURE_QUALIFICATION('', '#8389', (#8388));
#8392=MEASURE_QUALIFICATION('', '#8390', (#8388));
#8393=TOLERANCE_VALUE(#8390, #8389);
#8394=PLUS_MINUS_TOLERANCE(#8393, #8379);
#8395=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8379, #8387);
#8396=CARTESIAN_POINT('', (130.0, 0.0, 60.0));
#8397=DIRECTION('', (0.0, -1.0, 0.0));
#8398=AXIS1_PLACEMENT('', #8396, #8397);
#8399=SHAPE_ASPECT('', $, #38, .F.);
#8400=PROPERTY_DEFINITION('', $, #8399);
#8401=REPRESENTATION('', (#8398), #29);
#8402=PROPERTY_DEFINITION_REPRESENTATION(#8400, #8401);
#8403=DIMENSIONAL_LOCATION('', $, #143, #8399);
#8404=PRECISION_QUALIFIER(4);
#8410=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(120.0), #10) QUALIFIED_REPRESENTATION_ITEM((#8404)) REPRESENTA
TION_ITEM(''));
#8411=SHAPE_DIMENSION_REPRESENTATION('', (#8410), #29);
#8412=PRECISION_QUALIFIER(4);
#8413=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8414=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8415=MEASURE_QUALIFICATION('', '#8413', (#8412));
#8416=MEASURE_QUALIFICATION('', '#8414', (#8412));
#8417=TOLERANCE_VALUE(#8414, #8413);
#8418=PLUS_MINUS_TOLERANCE(#8417, #8403);
#8419=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8403, #8411);
#8425=(CHARACTERIZED_OBJECT('ROUND', $) FEATURE_DEFINITION() INSTANCED_FEATURE()
REMOVAL_VOLUME() SHAPE_ASPECT('ROUND', $, #38, .T.));
#8426=PROPERTY_DEFINITION('', $, #8425);
#8427=PRODUCT_DEFINITION_SHAPE('', $, #8425);
#8428=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#8429=DIRECTION('', (0.0, 0.0, 1.0));
#8430=DIRECTION('', (1.0, 0.0, 0.0));
#8431=AXIS2_PLACEMENT_3D('orientation', #8428, #8429, #8430);
#8432=REPRESENTATION('', (#4214, #4242, #4264, #4286, #4308, #4323, #4338, #4360, #438
8, #4410, #4432, #4454, #4469, #4484), #29);
#8433=PROPERTY_DEFINITION_REPRESENTATION(#8426, #8432);

```

```

#8434=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8431), #29);
#8435=PROPERTY_DEFINITION_REPRESENTATION(#8426, #8434);
#8436=SHAPE_ASPECT('', 'volume shape', #38, .T.);
#8437=PROPERTY_DEFINITION('', $, #8436);
#8438=PROPERTY_DEFINITION_REPRESENTATION(#8437, #8432);
#8439=SHAPE_ASPECT('', 'shape volume occurrence', #8427, .T.);
#8440=SHAPE_DEFINING_RELATIONSHIP('', 'volume shape usage', #8436, #8439);
#8446=(CHARACTERIZED_OBJECT('CUT', $) FEATURE_DEFINITION() INSTANCED_FEATURE() REMOVAL_VOLUME() SHAPE_ASPECT('CUT', $, #38, .T.));
#8447=PROPERTY_DEFINITION('', $, #8446);
#8448=PRODUCT_DEFINITION_SHAPE('', $, #8446);
#8449=CARTESIAN_POINT('', (0.0, 0.0, 0.0));
#8450=DIRECTION('', (0.0, 0.0, 1.0));
#8451=DIRECTION('', (1.0, 0.0, 0.0));
#8452=AXIS2_PLACEMENT_3D('orientation', #8449, #8450, #8451);
#8453=REPRESENTATION('', (#8452), #29);
#8454=PROPERTY_DEFINITION_REPRESENTATION(#8447, #8453);
#8455=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8452), #29);
#8456=PROPERTY_DEFINITION_REPRESENTATION(#8447, #8455);
#8457=SHAPE_ASPECT('', 'volume shape', #38, .T.);
#8458=PROPERTY_DEFINITION('', $, #8457);
#8459=PROPERTY_DEFINITION_REPRESENTATION(#8458, #8453);
#8460=SHAPE_ASPECT('', 'shape volume occurrence', #8448, .T.);
#8461=SHAPE_DEFINING_RELATIONSHIP('', 'volume shape usage', #8457, #8460);
#8467=(CHARACTERIZED_OBJECT('', $) FEATURE_DEFINITION() INSTANCED_FEATURE() ROUND_HOLE() SHAPE_ASPECT('', $, #38, .T.));
#8468=PROPERTY_DEFINITION('', $, #8467);
#8469=PRODUCT_DEFINITION_SHAPE('', $, #8467);
#8470=CARTESIAN_POINT('', (50.0, 0.0, 90.0));
#8471=DIRECTION('', (0.0, -1.0, 0.0));
#8472=DIRECTION('', (1.0, 0.0, 0.0));
#8473=AXIS2_PLACEMENT_3D('orientation', #8470, #8471, #8472);
#8474=REPRESENTATION('', (#4506), #29);
#8475=PROPERTY_DEFINITION_REPRESENTATION(#8468, #8474);
#8476=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8473), #29);
#8477=PROPERTY_DEFINITION_REPRESENTATION(#8468, #8476);
#8478=FEATURE_COMPONENT_DEFINITION('', $);
#8479=PRODUCT_DEFINITION_SHAPE('', $, #8478);
#8480=CIRCULAR_CLOSED_PROFILE('circular profile', $, #8479, .F.);
#8481=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.5000000000000000);
#8482=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.5000000000000000);
#8483=PRECISION_QUALIFIER(4);
#8489=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(18.0), #10) QUALIFIED_REPRESENTATION_ITEM((#8481, #8482, #8483) REPRESENTATION_ITEM('diameter'));
#8490=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#8473, #8489), #29);
#8491=PROPERTY_DEFINITION('', $, #8480);
#8492=PROPERTY_DEFINITION_REPRESENTATION(#8491, #8490);
#8493=FEATURE_COMPONENT_DEFINITION('', $);
#8494=PRODUCT_DEFINITION_SHAPE('', $, #8493);
#8495=PATH_FEATURE_COMPONENT('linear path', 'linear', #8494, .F.);
#8496=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#8497=STANDARD_UNCERTAINTY('lower limit', 'minus', -1.0);

```

ISO 10303-223:2008 (E)

```

#8498=PRECISION_QUALIFIER(4);
#8504=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(10.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8496,#8497,#8498)
)REPRESENTATION_ITEM('distance'));
#8505=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#8473,#8504),#29);
#8506=PROPERTY_DEFINITION('',$,#8495);
#8507=PROPERTY_DEFINITION_REPRESENTATION(#8506,#8505);
#8508=DIRECTION_SHAPE_REPRESENTATION('',(#8471),#29);
#8509=PROPERTY_DEFINITION_REPRESENTATION(#8506,#8508);
#8510=SHAPE_ASPECT('','diameter occurrence',#8469,.T.);
#8511=SHAPE_DEFINING_RELATIONSHIP('diameter','profile usage',#8480,#8510);
#8512=SHAPE_ASPECT('','hole depth occurrence',#8469,.T.);
#8513=SHAPE_DEFINING_RELATIONSHIP('hole depth','path feature component
usage',#8495,#8512);

#8514=FEATURE_COMPONENT_DEFINITION('',$);
#8515=PRODUCT_DEFINITION_SHAPE('',$,#8514);
#8516=HOLE_BOTTOM('bottom condition','flat',#8515,.F.);
#8521=SHAPE_ASPECT('','bottom condition occurrence',#8469,.T.);

#8522=FEATURE_COMPONENT_RELATIONSHIP('hole depth end','hole bottom
usage',#8516,#8521);
#8523=CARTESIAN_POINT('',(50.0,0.0,90.0));
#8524=DIRECTION('',(0.0,-1.0,0.0));
#8525=AXIS1_PLACEMENT('',#8523,#8524);
#8526=SHAPE_ASPECT('',$,#38,.F.);
#8527=PROPERTY_DEFINITION('',$,#8526);
#8528=REPRESENTATION('',(#8525),#29);
#8529=PROPERTY_DEFINITION_REPRESENTATION(#8527,#8528);
#8530=DIMENSIONAL_LOCATION('',$,#239,#8526);
#8531=PRECISION_QUALIFIER(4);
#8537=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(50.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8531))REPRESENTAT
ION_ITEM(''));
#8538=SHAPE_DIMENSION_REPRESENTATION('',(#8537),#29);
#8539=PRECISION_QUALIFIER(4);
#8540=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000),#10);
#8541=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0100000000000000),#10);
#8542=MEASURE_QUALIFICATION('',',',#8540,(#8539));
#8543=MEASURE_QUALIFICATION('',',',#8541,(#8539));
#8544=TOLERANCE_VALUE(#8541,#8540);
#8545=PLUS_MINUS_TOLERANCE(#8544,#8530);
#8546=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8530,#8538);
#8547=CARTESIAN_POINT('',(50.0,0.0,90.0));
#8548=DIRECTION('',(0.0,-1.0,0.0));
#8549=AXIS1_PLACEMENT('',#8547,#8548);
#8550=SHAPE_ASPECT('',$,#38,.F.);
#8551=PROPERTY_DEFINITION('',$,#8550);
#8552=REPRESENTATION('',(#8549),#29);
#8553=PROPERTY_DEFINITION_REPRESENTATION(#8551,#8552);
#8554=DIMENSIONAL_LOCATION('',$,#143,#8550);
#8555=PRECISION_QUALIFIER(4);

```



```

#8561=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(90.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8555))REPRESENTAT
ION_ITEM(''));
#8562=SHAPE_DIMENSION_REPRESENTATION('',( #8561),#29);
#8563=PRECISION_QUALIFIER(4);
#8564=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000),#10);
#8565=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.0100000000000000),#10);
#8566=MEASURE_QUALIFICATION('','',#8564,(#8563));
#8567=MEASURE_QUALIFICATION('','',#8565,(#8563));
#8568=TOLERANCE_VALUE(#8565,#8564);
#8569=PLUS_MINUS_TOLERANCE(#8568,#8554);
#8570=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8554,#8562);
#8576=(CHARACTERIZED_OBJECT('CUT',$)FEATURE_DEFINITION()INSTANCED_FEATURE()RE
MOVAL_VOLUME()SHAPE_ASPECT('CUT',$,#38,.T.));
#8577=PROPERTY_DEFINITION('',$,#8576);
#8578=PRODUCT_DEFINITION_SHAPE('',$,#8576);
#8579=CARTESIAN_POINT('',(0.0,0.0,0.0));
#8580=DIRECTION('',(0.0,0.0,1.0));
#8581=DIRECTION('',(1.0,0.0,0.0));
#8582=AXIS2_PLACEMENT_3D('orientation',#8579,#8580,#8581);
#8583=REPRESENTATION('',( #4622,#4642,#4662,#4682,#4702,#4718,#4742,#4762,#478
2,#4802,#4822,#4842,#4862,#4878),#29);
#8584=PROPERTY_DEFINITION_REPRESENTATION(#8577,#8583);
#8585=SHAPE_REPRESENTATION_WITH_PARAMETERS('',( #8582),#29);
#8586=PROPERTY_DEFINITION_REPRESENTATION(#8577,#8585);
#8587=SHAPE_ASPECT('','volume shape',#38,.T.);
#8588=PROPERTY_DEFINITION('',$,#8587);
#8589=PROPERTY_DEFINITION_REPRESENTATION(#8588,#8583);
#8590=SHAPE_ASPECT('','shape volume occurrence',#8578,.T.);
#8591=SHAPE_DEFINING_RELATIONSHIP('','volume shape usage',#8587,#8590);
#8592=DIMENSIONAL_SIZE(#4663,'radius');
#8593=PRECISION_QUALIFIER(4);
#8599=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8593))REPRESENTATI
ON_ITEM(''));
#8600=SHAPE_DIMENSION_REPRESENTATION('',( #8599),#29);
#8601=PRECISION_QUALIFIER(4);
#8602=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8603=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8604=MEASURE_QUALIFICATION('','',#8602,(#8601));
#8605=MEASURE_QUALIFICATION('','',#8603,(#8601));
#8606=TOLERANCE_VALUE(#8603,#8602);
#8607=PLUS_MINUS_TOLERANCE(#8606,#8592);
#8608=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8592,#8600);
#8609=DIMENSIONAL_SIZE(#4703,'radius');
#8610=PRECISION_QUALIFIER(4);
#8616=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8610))REPRESENTATI
ON_ITEM(''));
#8617=SHAPE_DIMENSION_REPRESENTATION('',( #8616),#29);
#8618=PRECISION_QUALIFIER(4);
#8619=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8620=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);

```

```

#8621=MEASURE_QUALIFICATION('', '#8619', (#8618));
#8622=MEASURE_QUALIFICATION('', '#8620', (#8618));
#8623=TOLERANCE_VALUE(#8620, #8619);
#8624=PLUS_MINUS_TOLERANCE(#8623, #8609);
#8625=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8609, #8617);
#8626=DIMENSIONAL_SIZE(#4879, 'radius');
#8627=PRECISION_QUALIFIER(4);
#8633=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8627))REPRESENTATI
ON_ITEM(''));
#8634=SHAPE_DIMENSION_REPRESENTATION('', (#8633), #29);
#8635=PRECISION_QUALIFIER(4);
#8636=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8637=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8638=MEASURE_QUALIFICATION('', '#8636', (#8635));
#8639=MEASURE_QUALIFICATION('', '#8637', (#8635));
#8640=TOLERANCE_VALUE(#8637, #8636);
#8641=PLUS_MINUS_TOLERANCE(#8640, #8626);
#8642=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8626, #8634);
#8643=DIMENSIONAL_SIZE(#4763, 'radius');
#8644=PRECISION_QUALIFIER(4);
#8650=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8644))REPRESENTATI
ON_ITEM(''));
#8651=SHAPE_DIMENSION_REPRESENTATION('', (#8650), #29);
#8652=PRECISION_QUALIFIER(4);
#8653=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8654=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8655=MEASURE_QUALIFICATION('', '#8653', (#8652));
#8656=MEASURE_QUALIFICATION('', '#8654', (#8652));
#8657=TOLERANCE_VALUE(#8654, #8653);
#8658=PLUS_MINUS_TOLERANCE(#8657, #8643);
#8659=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8643, #8651);
#8660=DIMENSIONAL_SIZE(#4803, 'radius');
#8661=PRECISION_QUALIFIER(4);
#8667=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8661))REPRESENTATI
ON_ITEM(''));
#8668=SHAPE_DIMENSION_REPRESENTATION('', (#8667), #29);
#8669=PRECISION_QUALIFIER(4);
#8670=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#8671=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#8672=MEASURE_QUALIFICATION('', '#8670', (#8669));
#8673=MEASURE_QUALIFICATION('', '#8671', (#8669));
#8674=TOLERANCE_VALUE(#8671, #8670);
#8675=PLUS_MINUS_TOLERANCE(#8674, #8660);
#8676=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8660, #8668);
#8677=DIMENSIONAL_SIZE(#4843, 'radius');
#8678=PRECISION_QUALIFIER(4);
#8684=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(2.0), #10)QUALIFIED_REPRESENTATION_ITEM((#8678))REPRESENTATI
ON_ITEM(''));
#8685=SHAPE_DIMENSION_REPRESENTATION('', (#8684), #29);

```

```

#8686=PRECISION_QUALIFIER(4);
#8687=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8688=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8689=MEASURE_QUALIFICATION('','',#8687,(#8686));
#8690=MEASURE_QUALIFICATION('','',#8688,(#8686));
#8691=TOLERANCE_VALUE(#8688,#8687);
#8692=PLUS_MINUS_TOLERANCE(#8691,#8677);
#8693=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8677,#8685);
#8694=CARTESIAN_POINT('',(47.9999999999999999,0.0,180.0));
#8695=DIRECTION('',(0.0,0.0,1.0));
#8696=DIRECTION('',(1.0,0.0,0.0));
#8697=AXIS2_PLACEMENT_3D('',#8694,#8695,#8696);
#8698=SHAPE_ASPECT('','$,#38,.F.);
#8699=PROPERTY_DEFINITION('','$,#8698);
#8700=REPRESENTATION('',( #8697),#29);
#8701=PROPERTY_DEFINITION_REPRESENTATION(#8699,#8700);
#8702=DIMENSIONAL_LOCATION('','$,#4743,#4823);
#8703=PRECISION_QUALIFIER(4);
#8709=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0000000000000007),#10)QUALIFIED_REPRESENTATION_ITEM((#870
3))REPRESENTATION_ITEM(''));
#8710=SHAPE_DIMENSION_REPRESENTATION('',( #8709),#29);
#8711=PRECISION_QUALIFIER(4);
#8712=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8713=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8714=MEASURE_QUALIFICATION('','',#8712,(#8711));
#8715=MEASURE_QUALIFICATION('','',#8713,(#8711));
#8716=TOLERANCE_VALUE(#8713,#8712);
#8717=PLUS_MINUS_TOLERANCE(#8716,#8702);
#8718=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8702,#8710);
#8719=DIMENSIONAL_LOCATION('','$,#4783,#4863);
#8720=PRECISION_QUALIFIER(4);
#8726=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(30.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8720))REPRESENTAT
ION_ITEM(''));
#8727=SHAPE_DIMENSION_REPRESENTATION('',( #8726),#29);
#8728=PRECISION_QUALIFIER(4);
#8729=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8730=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8731=MEASURE_QUALIFICATION('','',#8729,(#8728));
#8732=MEASURE_QUALIFICATION('','',#8730,(#8728));
#8733=TOLERANCE_VALUE(#8730,#8729);
#8734=PLUS_MINUS_TOLERANCE(#8733,#8719);
#8735=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8719,#8727);
#8736=CARTESIAN_POINT('',(47.9999999999999999,0.0,180.0));
#8737=DIRECTION('',(0.0,0.0,1.0));
#8738=DIRECTION('',(1.0,0.0,0.0));
#8739=AXIS2_PLACEMENT_3D('',#8736,#8737,#8738);
#8740=SHAPE_ASPECT('','$,#38,.F.);
#8741=PROPERTY_DEFINITION('','$,#8740);
#8742=REPRESENTATION('',( #8739),#29);
#8743=PROPERTY_DEFINITION_REPRESENTATION(#8741,#8742);
#8744=DIMENSIONAL_LOCATION('','$,#4683,#8740);

```

ISO 10303-223:2008 (E)

```
#8745=PRECISION_QUALIFIER(4);
#8751=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(49.99999999999993),#10)QUALIFIED_REPRESENTATION_ITEM((#874
5))REPRESENTATION_ITEM(''));
#8752=SHAPE_DIMENSION_REPRESENTATION('',( #8751),#29);
#8753=PRECISION_QUALIFIER(4);
#8754=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8755=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8756=MEASURE_QUALIFICATION('','',#8754,(#8753));
#8757=MEASURE_QUALIFICATION('','',#8755,(#8753));
#8758=TOLERANCE_VALUE(#8755,#8754);
#8759=PLUS_MINUS_TOLERANCE(#8758,#8744);
#8760=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8744,#8752);
#8761=CARTESIAN_POINT('',(47.999999999999979,0.0,180.0));
#8762=DIRECTION('',(0.0,0.0,1.0));
#8763=DIRECTION('',(1.0,0.0,0.0));
#8764=AXIS2_PLACEMENT_3D('','#8761,#8762,#8763);
#8765=SHAPE_ASPECT('','$,#38,.F.);
#8766=PROPERTY_DEFINITION('','$,#8765);
#8767=REPRESENTATION('',( #8764),#29);
#8768=PROPERTY_DEFINITION_REPRESENTATION(#8766,#8767);
#8769=DIMENSIONAL_LOCATION('','$,#8765,#4823);
#8770=PRECISION_QUALIFIER(4);
#8776=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.000000000000002),#10)QUALIFIED_REPRESENTATION_ITEM((#877
0))REPRESENTATION_ITEM(''));
#8777=SHAPE_DIMENSION_REPRESENTATION('',( #8776),#29);
#8778=PRECISION_QUALIFIER(4);
#8779=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8780=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8781=MEASURE_QUALIFICATION('','',#8779,(#8778));
#8782=MEASURE_QUALIFICATION('','',#8780,(#8778));
#8783=TOLERANCE_VALUE(#8780,#8779);
#8784=PLUS_MINUS_TOLERANCE(#8783,#8769);
#8785=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8769,#8777);
#8786=DIMENSIONAL_LOCATION('','$,#4643,#4783);
#8787=PRECISION_QUALIFIER(4);
#8793=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(19.999999999999972),#10)QUALIFIED_REPRESENTATION_ITEM((#878
7))REPRESENTATION_ITEM(''));
#8794=SHAPE_DIMENSION_REPRESENTATION('',( #8793),#29);
#8795=PRECISION_QUALIFIER(4);
#8796=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8797=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8798=MEASURE_QUALIFICATION('','',#8796,(#8795));
#8799=MEASURE_QUALIFICATION('','',#8797,(#8795));
#8800=TOLERANCE_VALUE(#8797,#8796);
#8801=PLUS_MINUS_TOLERANCE(#8800,#8786);
#8802=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8786,#8794);
#8803=DIMENSIONAL_LOCATION('','$,#5534,#4719);
#8804=PRECISION_QUALIFIER(4);
```



ISO 10303-223:2008 (E)

```

#8867=PRECISION_QUALIFIER(4);
#8873=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8867))REPRESENTATI
ON_ITEM(''));
#8874=SHAPE_DIMENSION_REPRESENTATION('',( #8873),#29);
#8875=PRECISION_QUALIFIER(4);
#8876=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8877=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8878=MEASURE_QUALIFICATION('','',#8876,(#8875));
#8879=MEASURE_QUALIFICATION('','',#8877,(#8875));
#8880=TOLERANCE_VALUE(#8877,#8876);
#8881=PLUS_MINUS_TOLERANCE(#8880,#8866);
#8882=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8866,#8874);
#8883=DIMENSIONAL_SIZE(#4923,'radius');
#8884=PRECISION_QUALIFIER(4);
#8890=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8884))REPRESENTATI
ON_ITEM(''));
#8891=SHAPE_DIMENSION_REPRESENTATION('',( #8890),#29);
#8892=PRECISION_QUALIFIER(4);
#8893=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8894=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8895=MEASURE_QUALIFICATION('','',#8893,(#8892));
#8896=MEASURE_QUALIFICATION('','',#8894,(#8892));
#8897=TOLERANCE_VALUE(#8894,#8893);
#8898=PLUS_MINUS_TOLERANCE(#8897,#8883);
#8899=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8883,#8891);
#8900=DIMENSIONAL_SIZE(#4987,'radius');
#8901=PRECISION_QUALIFIER(4);
#8907=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8901))REPRESENTATI
ON_ITEM(''));
#8908=SHAPE_DIMENSION_REPRESENTATION('',( #8907),#29);
#8909=PRECISION_QUALIFIER(4);
#8910=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8911=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8912=MEASURE_QUALIFICATION('','',#8910,(#8909));
#8913=MEASURE_QUALIFICATION('','',#8911,(#8909));
#8914=TOLERANCE_VALUE(#8911,#8910);
#8915=PLUS_MINUS_TOLERANCE(#8914,#8900);
#8916=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8900,#8908);
#8917=DIMENSIONAL_SIZE(#5115,'radius');
#8918=PRECISION_QUALIFIER(4);
#8924=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#8918))REPRESENTATI
ON_ITEM(''));
#8925=SHAPE_DIMENSION_REPRESENTATION('',( #8924),#29);
#8926=PRECISION_QUALIFIER(4);
#8927=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#8928=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000),#10);
#8929=MEASURE_QUALIFICATION('','',#8927,(#8926));
#8930=MEASURE_QUALIFICATION('','',#8928,(#8926));
#8931=TOLERANCE_VALUE(#8928,#8927);

```

```

#8932=PLUS_MINUS_TOLERANCE(#8931,#8917);
#8933=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8917,#8925);
#8934=CARTESIAN_POINT('',(228.0,-50.0,50.0));
#8935=DIRECTION('',(1.0,0.0,0.0));
#8936=DIRECTION('',(0.0,0.0,1.0));
#8937=AXIS2_PLACEMENT_3D('',#8934,#8935,#8936);
#8938=SHAPE_ASPECT('',$,#38,.F.);
#8939=PROPERTY_DEFINITION('',$,#8938);
#8940=REPRESENTATION('',(#8937),#29);
#8941=PROPERTY_DEFINITION_REPRESENTATION(#8939,#8940);
#8942=CARTESIAN_POINT('',(179.999999999999972,-50.0,178.0));
#8943=DIRECTION('',(0.0,0.0,1.0));
#8944=DIRECTION('',(-1.0,0.0,0.0));
#8945=AXIS2_PLACEMENT_3D('',#8942,#8943,#8944);
#8946=SHAPE_ASPECT('',$,#38,.F.);
#8947=PROPERTY_DEFINITION('',$,#8946);
#8948=REPRESENTATION('',(#8945),#29);
#8949=PROPERTY_DEFINITION_REPRESENTATION(#8947,#8948);
#8950=CARTESIAN_POINT('',(2.0,-50.0,129.999999999999972));
#8951=DIRECTION('',(-1.0,0.0,0.0));
#8952=DIRECTION('',(0.0,0.0,-1.0));
#8953=AXIS2_PLACEMENT_3D('',#8950,#8951,#8952);
#8954=SHAPE_ASPECT('',$,#38,.F.);
#8955=PROPERTY_DEFINITION('',$,#8954);
#8956=REPRESENTATION('',(#8953),#29);
#8957=PROPERTY_DEFINITION_REPRESENTATION(#8955,#8956);
#8958=CARTESIAN_POINT('',(50.0,-50.0,2.0));
#8959=DIRECTION('',(0.0,0.0,-1.0));
#8960=DIRECTION('',(1.0,0.0,0.0));
#8961=AXIS2_PLACEMENT_3D('',#8958,#8959,#8960);
#8962=SHAPE_ASPECT('',$,#38,.F.);
#8963=PROPERTY_DEFINITION('',$,#8962);
#8964=REPRESENTATION('',(#8961),#29);
#8965=PROPERTY_DEFINITION_REPRESENTATION(#8963,#8964);
#8966=DIMENSIONAL_LOCATION('',$,#5135,#5007);
#8967=PRECISION_QUALIFIER(4);
#8973=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(79.999999999999972),#10)QUALIFIED_REPRESENTATION_ITEM((#896
7)REPRESENTATION_ITEM(''));
#8974=SHAPE_DIMENSION_REPRESENTATION('',(#8973),#29);
#8975=PRECISION_QUALIFIER(4);
#8976=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.100000000000000),#10);
#8977=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.100000000000000),#10);
#8978=MEASURE_QUALIFICATION('','',#8976,(#8975));
#8979=MEASURE_QUALIFICATION('','',#8977,(#8975));
#8980=TOLERANCE_VALUE(#8977,#8976);
#8981=PLUS_MINUS_TOLERANCE(#8980,#8966);
#8982=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8966,#8974);
#8983=CARTESIAN_POINT('',(228.0,-50.0,50.0));
#8984=DIRECTION('',(1.0,0.0,0.0));
#8985=DIRECTION('',(0.0,0.0,1.0));
#8986=AXIS2_PLACEMENT_3D('',#8983,#8984,#8985);
#8987=SHAPE_ASPECT('',$,#38,.F.);

```

```

#8988=PROPERTY_DEFINITION('', $, #8987);
#8989=REPRESENTATION('', (#8986), #29);
#8990=PROPERTY_DEFINITION_REPRESENTATION(#8988, #8989);
#8991=DIMENSIONAL_LOCATION('', $, #5095, #8987);
#8992=PRECISION_QUALIFIER(4);
#8998=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(178.000000000000028), #10)QUALIFIED_REPRESENTATION_ITEM((#89
92))REPRESENTATION_ITEM(''));
#8999=SHAPE_DIMENSION_REPRESENTATION('', (#8998), #29);
#9000=PRECISION_QUALIFIER(4);
#9001=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#9002=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#9003=MEASURE_QUALIFICATION('', '#9001, (#9000));
#9004=MEASURE_QUALIFICATION('', '#9002, (#9000));
#9005=TOLERANCE_VALUE(#9002, #9001);
#9006=PLUS_MINUS_TOLERANCE(#9005, #8991);
#9007=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#8991, #8999);
#9008=DIMENSIONAL_LOCATION('', $, #2264, #4903);
#9009=PRECISION_QUALIFIER(4);
#9015=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(48.000000000000007), #10)QUALIFIED_REPRESENTATION_ITEM((#900
9))REPRESENTATION_ITEM(''));
#9016=SHAPE_DIMENSION_REPRESENTATION('', (#9015), #29);
#9017=PRECISION_QUALIFIER(4);
#9018=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#9019=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#9020=MEASURE_QUALIFICATION('', '#9018, (#9017));
#9021=MEASURE_QUALIFICATION('', '#9019, (#9017));
#9022=TOLERANCE_VALUE(#9019, #9018);
#9023=PLUS_MINUS_TOLERANCE(#9022, #9008);
#9024=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9008, #9016);
#9025=DIMENSIONAL_LOCATION('', $, #2178, #4943);
#9026=PRECISION_QUALIFIER(4);
#9032=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(48.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9026))REPRESENTAT
ION_ITEM(''));
#9033=SHAPE_DIMENSION_REPRESENTATION('', (#9032), #29);
#9034=PRECISION_QUALIFIER(4);
#9035=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#9036=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#9037=MEASURE_QUALIFICATION('', '#9035, (#9034));
#9038=MEASURE_QUALIFICATION('', '#9036, (#9034));
#9039=TOLERANCE_VALUE(#9036, #9035);
#9040=PLUS_MINUS_TOLERANCE(#9039, #9025);
#9041=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9025, #9033);
#9042=CARTESIAN_POINT('', (0.0, -40.0, 0.0));
#9043=DIRECTION('', (0.0, -1.0, 0.0));
#9044=DIRECTION('', (1.0, 0.0, 0.0));
#9045=AXIS2_PLACEMENT_3D('', #9042, #9044);
#9046=SHAPE_ASPECT('', $, #38, .F.);
#9047=PROPERTY_DEFINITION('', $, #9046);
#9048=REPRESENTATION('', (#9045), #29);
#9049=PROPERTY_DEFINITION_REPRESENTATION(#9047, #9048);

```



```

#9050=CARTESIAN_POINT('', (0.0, -50.0, 0.0));
#9051=DIRECTION('', (0.0, -1.0, 0.0));
#9052=DIRECTION('', (1.0, 0.0, 0.0));
#9053=AXIS2_PLACEMENT_3D('', #9050, #9051, #9052);
#9054=SHAPE_ASPECT('', $, #38, .F.);
#9055=PROPERTY_DEFINITION('', $, #9054);
#9056=REPRESENTATION('', (#9053), #29);
#9057=PROPERTY_DEFINITION_REPRESENTATION(#9055, #9056);
#9058=DIMENSIONAL_LOCATION('', $, #9046, #9054);
#9059=PRECISION_QUALIFIER(4);
#9065=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(10.0), #10) QUALIFIED_REPRESENTATION_ITEM((#9059)) REPRESENTATION_ITEM(''));
#9066=SHAPE_DIMENSION_REPRESENTATION('', (#9065), #29);
#9067=PRECISION_QUALIFIER(4);
#9068=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#9069=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#9070=MEASURE_QUALIFICATION('', '#9068', (#9067));
#9071=MEASURE_QUALIFICATION('', '#9069', (#9067));
#9072=TOLERANCE_VALUE(#9069, #9068);
#9073=PLUS_MINUS_TOLERANCE(#9072, #9058);
#9074=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9058, #9066);
#9080=(CHARACTERIZED_OBJECT('', 'open rectangular') FEATURE_DEFINITION() INSTANCED_FEATURE() POCKET() SHAPE_ASPECT('', 'open rectangular', #38, .T.));
#9081=PROPERTY_DEFINITION('', $, #9080);
#9082=PRODUCT_DEFINITION_SHAPE('', $, #9080);
#9083=CARTESIAN_POINT('', (66.0, -10.0, 90.0));
#9084=DIRECTION('', (0.0, -1.0, 0.0));
#9085=DIRECTION('', (0.0, 0.0, 1.0));
#9086=AXIS2_PLACEMENT_3D('orientation', #9083, #9084, #9085);
#9087=REPRESENTATION('', (#5316, #5366, #5386, #5406, #5426, #5446, #5462), #29);
#9088=PROPERTY_DEFINITION_REPRESENTATION(#9081, #9087);
#9089=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9086), #29);
#9090=PROPERTY_DEFINITION_REPRESENTATION(#9081, #9089);
#9091=FEATURE_COMPONENT_DEFINITION('', $);
#9092=PRODUCT_DEFINITION_SHAPE('', $, #9091);
#9093=SQUARE_U_PROFILE('square u profile', 'boundary', #9092, .F.);
#9094=STANDARD_UNCERTAINTY('upper limit', 'plus', 1.0);
#9095=STANDARD_UNCERTAINTY('lower limit', 'minus', -1.0);
#9096=PRECISION_QUALIFIER(4);
#9102=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(LENGTH_MEASURE(32.0), #10) QUALIFIED_REPRESENTATION_ITEM((#9094, #9095, #9096)) REPRESENTATION_ITEM('width'));
#9107=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('first angle'));
#9112=(MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(90.0), #19) PLANE_ANGLE_MEASURE_WITH_UNIT() REPRESENTATION_ITEM('second angle'));
#9113=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.5000000000000000);
#9114=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.5000000000000000);
#9115=PRECISION_QUALIFIER(4);

```

```

#9121=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.900000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#911
3,#9114,#9115))REPRESENTATION_ITEM('first radius'));
#9122=STANDARD_UNCERTAINTY('upper limit','plus',0.5000000000000000);
#9123=STANDARD_UNCERTAINTY('lower limit','minus',-0.5000000000000000);
#9124=PRECISION_QUALIFIER(4);
#9130=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(15.900000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#912
2,#9123,#9124))REPRESENTATION_ITEM('second radius'));
#9131=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9086,#9102,#9107,#9112,#9121,
#9130),#29);
#9132=PROPERTY_DEFINITION('',$,#9093);
#9133=PROPERTY_DEFINITION_REPRESENTATION(#9132,#9131);
#9134=FEATURE_COMPONENT_DEFINITION('',$);
#9135=PRODUCT_DEFINITION_SHAPE('',$,#9134);
#9136=PATH_FEATURE_COMPONENT('linear path','linear',#9135,.F.);
#9137=STANDARD_UNCERTAINTY('upper limit','plus',1.0);
#9138=STANDARD_UNCERTAINTY('lower limit','minus',-1.0);
#9139=PRECISION_QUALIFIER(4);
#9145=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(20.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9137,#9138,#9139)
)REPRESENTATION_ITEM('distance'));
#9146=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9086,#9145),#29);
#9147=PROPERTY_DEFINITION('',$,#9136);
#9148=PROPERTY_DEFINITION_REPRESENTATION(#9147,#9146);
#9149=DIRECTION_SHAPE_REPRESENTATION('',(#9084),#29);
#9150=PROPERTY_DEFINITION_REPRESENTATION(#9147,#9149);
#9151=SHAPE_ASPECT('','open boundary occurrence',#9082,.T.);
#9152=SHAPE_DEFINING_RELATIONSHIP('','profile usage',#9093,#9151);
#9153=SHAPE_ASPECT('','pocket depth occurrence',#9082,.T.);
#9154=SHAPE_DEFINING_RELATIONSHIP('pocket depth','path feature component
usage',#9136,#9153);
#9155=FEATURE_COMPONENT_DEFINITION('',$);
#9156=PRODUCT_DEFINITION_SHAPE('',$,#9155);
#9157=POCKET_BOTTOM('bottom condition','planar',#9156,.F.);
#9158=DESCRIPTIVE_REPRESENTATION_ITEM('pocket bottom orientation','pocket
depth end');
#9159=PROPERTY_DEFINITION('',$,#9157);
#9160=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9158),#29);
#9161=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9160);
#9162=DIRECTION('',(0.0,1.0,0.0));
#9163=DIRECTION_SHAPE_REPRESENTATION('floor normal',(#9162),#29);
#9164=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9163);
#9165=CARTESIAN_POINT('',(66.0,-30.0,90.0));
#9166=LOCATION_SHAPE_REPRESENTATION('floor location',(#9165),#29);
#9167=PROPERTY_DEFINITION_REPRESENTATION(#9159,#9166);
#9168=SHAPE_ASPECT('','bottom condition occurrence',#9082,.T.);
#9169=FEATURE_COMPONENT_RELATIONSHIP('','pocket bottom usage',#9157,#9168);
#9170=DIMENSIONAL_LOCATION('',$,#75,#5463);
#9171=PRECISION_QUALIFIER(4);
#9177=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(74.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9171))REPRESENTAT
ION_ITEM(''));

```

```

#9178=SHAPE_DIMENSION_REPRESENTATION('', (#9177), #29);
#9179=PRECISION_QUALIFIER(4);
#9180=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000), #10);
#9181=MEASURE_WITH_UNIT(LENGTH_MEASURE(-0.1000000000000000), #10);
#9182=MEASURE_QUALIFICATION('', '#9180', (#9179));
#9183=MEASURE_QUALIFICATION('', '#9181', (#9179));
#9184=TOLERANCE_VALUE(#9181, #9180);
#9185=PLUS_MINUS_TOLERANCE(#9184, #9170);
#9186=DIMENSIONAL_CHARACTERISTIC_REPRESENTATION(#9170, #9178);

#9187=EDGE_ROUND('', 'constant radius', #38, .T.);
#9188=PROPERTY_DEFINITION('', $, #9187);
#9189=FACE_SHAPE_REPRESENTATION('edge round face', (#3724), #29);
#9190=PROPERTY_DEFINITION_REPRESENTATION(#9188, #9189);
#9191=FACE_SHAPE_REPRESENTATION('first face shape', (#1260), #29);
#9192=PROPERTY_DEFINITION_REPRESENTATION(#9188, #9191);
#9193=FACE_SHAPE_REPRESENTATION('second face shape', (#74), #29);
#9194=PROPERTY_DEFINITION_REPRESENTATION(#9188, #9193);
#9195=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9196=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9197=PRECISION_QUALIFIER(4);
#9203=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9195, #9196, #9197))
REPRESENTATION_ITEM('radius'));
#9208=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9213=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9214=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9203, #9208, #9213), #29);
#9215=PROPERTY_DEFINITION_REPRESENTATION(#9188, #9214);

#9220=EDGE_ROUND('', 'constant radius', #38, .T.);
#9221=PROPERTY_DEFINITION('', $, #9220);
#9222=FACE_SHAPE_REPRESENTATION('edge round face', (#3740), #29);
#9223=PROPERTY_DEFINITION_REPRESENTATION(#9221, #9222);
#9224=FACE_SHAPE_REPRESENTATION('first face shape', (#1260), #29);
#9225=PROPERTY_DEFINITION_REPRESENTATION(#9221, #9224);
#9226=FACE_SHAPE_REPRESENTATION('second face shape', (#142), #29);
#9227=PROPERTY_DEFINITION_REPRESENTATION(#9221, #9226);
#9228=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9229=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9230=PRECISION_QUALIFIER(4);
#9236=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9228, #9229, #9230))
REPRESENTATION_ITEM('radius'));
#9241=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9246=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9247=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9236, #9241, #9246), #29);
#9248=PROPERTY_DEFINITION_REPRESENTATION(#9221, #9247);

```

```

#9253=EDGE_ROUND('', 'constant radius', #38, .T.);
#9254=PROPERTY_DEFINITION('', $, #9253);
#9255=FACE_SHAPE_REPRESENTATION('edge round face', (#3756), #29);
#9256=PROPERTY_DEFINITION_REPRESENTATION(#9254, #9255);
#9257=FACE_SHAPE_REPRESENTATION('first face shape', (#142), #29);
#9258=PROPERTY_DEFINITION_REPRESENTATION(#9254, #9257);
#9259=FACE_SHAPE_REPRESENTATION('second face shape', (#238), #29);
#9260=PROPERTY_DEFINITION_REPRESENTATION(#9254, #9259);
#9261=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9262=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9263=PRECISION_QUALIFIER(4);
#9269=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9261, #9262, #9263))
REPRESENTATION_ITEM('radius'));
#9274=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9279=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9280=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9269, #9274, #9279), #29);
#9281=PROPERTY_DEFINITION_REPRESENTATION(#9254, #9280);

#9286=EDGE_ROUND('', 'constant radius', #38, .T.);
#9287=PROPERTY_DEFINITION('', $, #9286);
#9288=FACE_SHAPE_REPRESENTATION('edge round face', (#3772), #29);
#9289=PROPERTY_DEFINITION_REPRESENTATION(#9287, #9288);
#9290=FACE_SHAPE_REPRESENTATION('first face shape', (#74), #29);
#9291=PROPERTY_DEFINITION_REPRESENTATION(#9287, #9290);
#9292=FACE_SHAPE_REPRESENTATION('second face shape', (#238), #29);
#9293=PROPERTY_DEFINITION_REPRESENTATION(#9287, #9292);
#9294=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9295=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9296=PRECISION_QUALIFIER(4);
#9302=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9294, #9295, #9296))
REPRESENTATION_ITEM('radius'));
#9307=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9312=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9313=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9302, #9307, #9312), #29);
#9314=PROPERTY_DEFINITION_REPRESENTATION(#9287, #9313);

#9319=EDGE_ROUND('', 'constant radius', #38, .T.);
#9320=PROPERTY_DEFINITION('', $, #9319);
#9321=FACE_SHAPE_REPRESENTATION('edge round face', (#3788), #29);
#9322=PROPERTY_DEFINITION_REPRESENTATION(#9320, #9321);
#9323=FACE_SHAPE_REPRESENTATION('first face shape', (#647), #29);
#9324=PROPERTY_DEFINITION_REPRESENTATION(#9320, #9323);
#9325=FACE_SHAPE_REPRESENTATION('second face shape', (#1876), #29);
#9326=PROPERTY_DEFINITION_REPRESENTATION(#9320, #9325);
#9327=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9328=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);

```

```

#9329=PRECISION_QUALIFIER(4);
#9335=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9327,#9328,#9329)
REPRESENTATION_ITEM('radius'));
#9340=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9345=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9346=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9335,#9340,#9345),#29);
#9347=PROPERTY_DEFINITION_REPRESENTATION(#9320,#9346);

#9352=EDGE_ROUND('', 'constant radius',#38,.T.);
#9353=PROPERTY_DEFINITION('', $, #9352);
#9354=FACE_SHAPE_REPRESENTATION('edge round face', (#3804), #29);
#9355=PROPERTY_DEFINITION_REPRESENTATION(#9353, #9354);
#9356=FACE_SHAPE_REPRESENTATION('first face shape', (#647), #29);
#9357=PROPERTY_DEFINITION_REPRESENTATION(#9353, #9356);
#9358=FACE_SHAPE_REPRESENTATION('second face shape', (#1836), #29);
#9359=PROPERTY_DEFINITION_REPRESENTATION(#9353, #9358);
#9360=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9361=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9362=PRECISION_QUALIFIER(4);
#9368=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9360,#9361,#9362)
REPRESENTATION_ITEM('radius'));
#9373=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9378=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9379=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9368,#9373,#9378), #29);
#9380=PROPERTY_DEFINITION_REPRESENTATION(#9353, #9379);

#9385=EDGE_ROUND('', 'constant radius',#38,.T.);
#9386=PROPERTY_DEFINITION('', $, #9385);
#9387=FACE_SHAPE_REPRESENTATION('edge round face', (#3820), #29);
#9388=PROPERTY_DEFINITION_REPRESENTATION(#9386, #9387);
#9389=FACE_SHAPE_REPRESENTATION('first face shape', (#2153), #29);
#9390=PROPERTY_DEFINITION_REPRESENTATION(#9386, #9389);
#9391=FACE_SHAPE_REPRESENTATION('second face shape', (#2220), #29);
#9392=PROPERTY_DEFINITION_REPRESENTATION(#9386, #9391);
#9393=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9394=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9395=PRECISION_QUALIFIER(4);
#9401=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9393,#9394,#9395)
REPRESENTATION_ITEM('radius'));
#9406=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9411=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9412=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9401,#9406,#9411), #29);
#9413=PROPERTY_DEFINITION_REPRESENTATION(#9386, #9412);

```

ISO 10303-223:2008 (E)

```

#9418=EDGE_ROUND('', 'constant radius', #38, .T.);
#9419=PROPERTY_DEFINITION('', $, #9418);
#9420=FACE_SHAPE_REPRESENTATION('edge round face', (#3836), #29);
#9421=PROPERTY_DEFINITION_REPRESENTATION(#9419, #9420);
#9422=FACE_SHAPE_REPRESENTATION('first face shape', (#2134), #29);
#9423=PROPERTY_DEFINITION_REPRESENTATION(#9419, #9422);
#9424=FACE_SHAPE_REPRESENTATION('second face shape', (#2282), #29);
#9425=PROPERTY_DEFINITION_REPRESENTATION(#9419, #9424);
#9426=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9427=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9428=PRECISION_QUALIFIER(4);
#9434=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) QUALIFIED_REPRESENTATION_ITEM((#9426, #9427, #9428))
REPRESENTATION_ITEM('radius'));
#9439=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) REPRESENTATION_ITEM('first offset'));
#9444=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) REPRESENTATION_ITEM('second offset'));
#9445=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9434, #9439, #9444), #29);
#9446=PROPERTY_DEFINITION_REPRESENTATION(#9419, #9445);

#9451=EDGE_ROUND('', 'constant radius', #38, .T.);
#9452=PROPERTY_DEFINITION('', $, #9451);
#9453=FACE_SHAPE_REPRESENTATION('edge round face', (#3852), #29);
#9454=PROPERTY_DEFINITION_REPRESENTATION(#9452, #9453);
#9455=FACE_SHAPE_REPRESENTATION('first face shape', (#2196), #29);
#9456=PROPERTY_DEFINITION_REPRESENTATION(#9452, #9455);
#9457=FACE_SHAPE_REPRESENTATION('second face shape', (#2239), #29);
#9458=PROPERTY_DEFINITION_REPRESENTATION(#9452, #9457);
#9459=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9460=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9461=PRECISION_QUALIFIER(4);
#9467=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) QUALIFIED_REPRESENTATION_ITEM((#9459, #9460, #9461))
REPRESENTATION_ITEM('radius'));
#9472=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) REPRESENTATION_ITEM('first offset'));
#9477=(LENGTH_MEASURE_WITH_UNIT() MEASURE_REPRESENTATION_ITEM() MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10) REPRESENTATION_ITEM('second offset'));
#9478=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9467, #9472, #9477), #29);
#9479=PROPERTY_DEFINITION_REPRESENTATION(#9452, #9478);

#9484=EDGE_ROUND('', 'constant radius', #38, .T.);
#9485=PROPERTY_DEFINITION('', $, #9484);
#9486=FACE_SHAPE_REPRESENTATION('edge round face', (#3868), #29);
#9487=PROPERTY_DEFINITION_REPRESENTATION(#9485, #9486);
#9488=FACE_SHAPE_REPRESENTATION('first face shape', (#2177), #29);
#9489=PROPERTY_DEFINITION_REPRESENTATION(#9485, #9488);
#9490=FACE_SHAPE_REPRESENTATION('second face shape', (#2263), #29);
#9491=PROPERTY_DEFINITION_REPRESENTATION(#9485, #9490);
#9492=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.1000000000000000);
#9493=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.1000000000000000);
#9494=PRECISION_QUALIFIER(4);

```

```

#9500=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9492,#9493,#9494)
REPRESENTATION_ITEM('radius'));
#9505=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9510=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9511=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9500,#9505,#9510),#29);
#9512=PROPERTY_DEFINITION_REPRESENTATION(#9485,#9511);

#9517=EDGE_ROUND('', 'constant radius', #38, .T.);
#9518=PROPERTY_DEFINITION('', $, #9517);
#9519=FACE_SHAPE_REPRESENTATION('edge round face', (#5150), #29);
#9520=PROPERTY_DEFINITION_REPRESENTATION(#9518, #9519);
#9521=FACE_SHAPE_REPRESENTATION('first face shape', (#2282), #29);
#9522=PROPERTY_DEFINITION_REPRESENTATION(#9518, #9521);
#9523=FACE_SHAPE_REPRESENTATION('second face shape', (#4966), #29);
#9524=PROPERTY_DEFINITION_REPRESENTATION(#9518, #9523);
#9525=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9526=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9527=PRECISION_QUALIFIER(4);
#9533=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9525,#9526,#9527)
REPRESENTATION_ITEM('radius'));
#9538=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9543=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9544=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9533,#9538,#9543), #29);
#9545=PROPERTY_DEFINITION_REPRESENTATION(#9518, #9544);

#9550=EDGE_ROUND('', 'constant radius', #38, .T.);
#9551=PROPERTY_DEFINITION('', $, #9550);
#9552=FACE_SHAPE_REPRESENTATION('edge round face', (#5166), #29);
#9553=PROPERTY_DEFINITION_REPRESENTATION(#9551, #9552);
#9554=FACE_SHAPE_REPRESENTATION('first face shape', (#2220), #29);
#9555=PROPERTY_DEFINITION_REPRESENTATION(#9551, #9554);
#9556=FACE_SHAPE_REPRESENTATION('second face shape', (#5070), #29);
#9557=PROPERTY_DEFINITION_REPRESENTATION(#9551, #9556);
#9558=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9559=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9560=PRECISION_QUALIFIER(4);
#9566=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9558,#9559,#9560)
REPRESENTATION_ITEM('radius'));
#9571=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9576=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9577=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9566,#9571,#9576), #29);
#9578=PROPERTY_DEFINITION_REPRESENTATION(#9551, #9577);

#9583=EDGE_ROUND('', 'constant radius', #38, .T.);

```

```

#9584=PROPERTY_DEFINITION('', $, #9583);
#9585=FACE_SHAPE_REPRESENTATION('edge round face', (#5182), #29);
#9586=PROPERTY_DEFINITION_REPRESENTATION(#9584, #9585);
#9587=FACE_SHAPE_REPRESENTATION('first face shape', (#2177), #29);
#9588=PROPERTY_DEFINITION_REPRESENTATION(#9584, #9587);
#9589=FACE_SHAPE_REPRESENTATION('second face shape', (#4902), #29);
#9590=PROPERTY_DEFINITION_REPRESENTATION(#9584, #9589);
#9591=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9592=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9593=PRECISION_QUALIFIER(4);
#9599=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9591, #9592, #9593))
REPRESENTATION_ITEM('radius'));
#9604=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9609=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9610=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9599, #9604, #9609), #29);
#9611=PROPERTY_DEFINITION_REPRESENTATION(#9584, #9610);

#9616=EDGE_ROUND('', 'constant radius', #38, .T.);
#9617=PROPERTY_DEFINITION('', $, #9616);
#9618=FACE_SHAPE_REPRESENTATION('edge round face', (#5198), #29);
#9619=PROPERTY_DEFINITION_REPRESENTATION(#9617, #9618);
#9620=FACE_SHAPE_REPRESENTATION('first face shape', (#2134), #29);
#9621=PROPERTY_DEFINITION_REPRESENTATION(#9617, #9620);
#9622=FACE_SHAPE_REPRESENTATION('second face shape', (#5006), #29);
#9623=PROPERTY_DEFINITION_REPRESENTATION(#9617, #9622);
#9624=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9625=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9626=PRECISION_QUALIFIER(4);
#9632=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9624, #9625, #9626))
REPRESENTATION_ITEM('radius'));
#9637=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9642=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9643=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9632, #9637, #9642), #29);
#9644=PROPERTY_DEFINITION_REPRESENTATION(#9617, #9643);

#9649=EDGE_ROUND('', 'constant radius', #38, .T.);
#9650=PROPERTY_DEFINITION('', $, #9649);
#9651=FACE_SHAPE_REPRESENTATION('edge round face', (#5214), #29);
#9652=PROPERTY_DEFINITION_REPRESENTATION(#9650, #9651);
#9653=FACE_SHAPE_REPRESENTATION('first face shape', (#2153), #29);
#9654=PROPERTY_DEFINITION_REPRESENTATION(#9650, #9653);
#9655=FACE_SHAPE_REPRESENTATION('second face shape', (#5030), #29);
#9656=PROPERTY_DEFINITION_REPRESENTATION(#9650, #9655);
#9657=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9658=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9659=PRECISION_QUALIFIER(4);

```



```

#9665=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9657,#9658,#9659))
REPRESENTATION_ITEM('radius'));
#9670=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9675=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9676=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9665,#9670,#9675),#29);
#9677=PROPERTY_DEFINITION_REPRESENTATION(#9650,#9676);

#9682=EDGE_ROUND('', 'constant radius', #38, .T.);
#9683=PROPERTY_DEFINITION('', $, #9682);
#9684=FACE_SHAPE_REPRESENTATION('edge round face', (#5230), #29);
#9685=PROPERTY_DEFINITION_REPRESENTATION(#9683, #9684);
#9686=FACE_SHAPE_REPRESENTATION('first face shape', (#2263), #29);
#9687=PROPERTY_DEFINITION_REPRESENTATION(#9683, #9686);
#9688=FACE_SHAPE_REPRESENTATION('second face shape', (#4942), #29);
#9689=PROPERTY_DEFINITION_REPRESENTATION(#9683, #9688);
#9690=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9691=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9692=PRECISION_QUALIFIER(4);
#9698=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9690,#9691,#9692))
REPRESENTATION_ITEM('radius'));
#9703=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9708=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9709=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9698, #9703, #9708), #29);
#9710=PROPERTY_DEFINITION_REPRESENTATION(#9683, #9709);

#9715=EDGE_ROUND('', 'constant radius', #38, .T.);
#9716=PROPERTY_DEFINITION('', $, #9715);
#9717=FACE_SHAPE_REPRESENTATION('edge round face', (#5246), #29);
#9718=PROPERTY_DEFINITION_REPRESENTATION(#9716, #9717);
#9719=FACE_SHAPE_REPRESENTATION('first face shape', (#2239), #29);
#9720=PROPERTY_DEFINITION_REPRESENTATION(#9716, #9719);
#9721=FACE_SHAPE_REPRESENTATION('second face shape', (#5094), #29);
#9722=PROPERTY_DEFINITION_REPRESENTATION(#9716, #9721);
#9723=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9724=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9725=PRECISION_QUALIFIER(4);
#9731=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)QUALIFIED_REPRESENTATION_ITEM((#9723,#9724,#9725))
REPRESENTATION_ITEM('radius'));
#9736=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('first offset'));
#9741=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0),#10)REPRESENTATION_ITEM('second offset'));
#9742=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9731, #9736, #9741), #29);
#9743=PROPERTY_DEFINITION_REPRESENTATION(#9716, #9742);

#9748=EDGE_ROUND('', 'constant radius', #38, .T.);

```

```

#9749=PROPERTY_DEFINITION('', $, #9748);
#9750=FACE_SHAPE_REPRESENTATION('edge round face', (#5262), #29);
#9751=PROPERTY_DEFINITION_REPRESENTATION(#9749, #9750);
#9752=FACE_SHAPE_REPRESENTATION('first face shape', (#2196), #29);
#9753=PROPERTY_DEFINITION_REPRESENTATION(#9749, #9752);
#9754=FACE_SHAPE_REPRESENTATION('second face shape', (#5134), #29);
#9755=PROPERTY_DEFINITION_REPRESENTATION(#9749, #9754);
#9756=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0100000000000000);
#9757=STANDARD_UNCERTAINTY('lower limit', 'minus', -0.0100000000000000);
#9758=PRECISION_QUALIFIER(4);
#9764=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)QUALIFIED_REPRESENTATION_ITEM((#9756, #9757, #9758)
REPRESENTATION_ITEM('radius'));
#9769=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('first offset'));
#9774=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(5.0), #10)REPRESENTATION_ITEM('second offset'));
#9775=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9764, #9769, #9774), #29);
#9776=PROPERTY_DEFINITION_REPRESENTATION(#9749, #9775);

#9786=(CHARACTERIZED_OBJECT('', 'thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('', 'thread', #38, .T.)THREAD());
#9787=PROPERTY_DEFINITION('', $, #9786);
#9788=CARTESIAN_POINT('', (190.0, 0.0, 120.0));
#9789=DIRECTION('', (0.0, -1.0, 0.0));
#9790=DIRECTION('', (1.0, 0.0, 0.0));
#9791=AXIS2_PLACEMENT_3D('orientation', #9788, #9789, #9790);
#9792=DESCRIPTIVE_REPRESENTATION_ITEM('thread side', 'internal');
#9793=DESCRIPTIVE_REPRESENTATION_ITEM('form', 'UNC');
#9794=DESCRIPTIVE_REPRESENTATION_ITEM('fit class', '2B');
#9795=DESCRIPTIVE_REPRESENTATION_ITEM('hand', 'right');
#9796=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9797=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0024000000000000);
#9798=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0024000000000000);
#9799=PRECISION_QUALIFIER(4);
#9805=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2199000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9797
, #9798, #9799)REPRESENTATION_ITEM('pitch diameter'));
#9810=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000), #10)REPRESENTATION_ITEM('major
diameter'));
#9811=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0055000000000000);
#9812=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0055000000000000);
#9813=PRECISION_QUALIFIER(4);
#9819=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2015000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9811
, #9812, #9813)REPRESENTATION_ITEM('minor diameter'));
#9820=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9791, #9792, #9793, #9794, #9795,
#9796, #9805, #9810, #9819), #29);
#9821=PROPERTY_DEFINITION_REPRESENTATION(#9787, #9820);
#9822=SHAPE_DEFINING_RELATIONSHIP('', 'applied shape', #3499, #9833);

```

```

#9823=APPLIED_AREA('', $, #38, .T.);
#9824=PROPERTY_DEFINITION('', $, #9823);
#9829=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999), #10)REPRESENTATION_ITEM('length'));
#9830=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9791, #9829), #29);
#9831=PROPERTY_DEFINITION_REPRESENTATION(#9824, #9830);
#9832=PRODUCT_DEFINITION_SHAPE('', $, #9786);
#9833=SHAPE_ASPECT('', 'partial area occurrence', #9832, .T.);
#9834=SHAPE_DEFINING_RELATIONSHIP('', 'applied area usage', #9823, #9833);

#9840=(CHARACTERIZED_OBJECT('', 'thread')FEATURE_DEFINITION()INSTANCED_FEATURE
()SHAPE_ASPECT('', 'thread', #38, .T.)THREAD());
#9841=PROPERTY_DEFINITION('', $, #9840);
#9842=CARTESIAN_POINT('', (190.0, 0.0, 60.0));
#9843=DIRECTION('', (0.0, -1.0, 0.0));
#9844=DIRECTION('', (1.0, 0.0, 0.0));
#9845=AXIS2_PLACEMENT_3D('orientation', #9842, #9843, #9844);
#9846=DESCRIPTIVE_REPRESENTATION_ITEM('thread side', 'internal');
#9847=DESCRIPTIVE_REPRESENTATION_ITEM('form', 'UNC');
#9848=DESCRIPTIVE_REPRESENTATION_ITEM('fit class', '2B');
#9849=DESCRIPTIVE_REPRESENTATION_ITEM('hand', 'right');
#9850=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9851=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0024000000000000);
#9852=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0024000000000000);
#9853=PRECISION_QUALIFIER(4);
#9859=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2199000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9851
, #9852, #9853))REPRESENTATION_ITEM('pitch diameter'));
#9864=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000), #10)REPRESENTATION_ITEM('major
diameter'));
#9865=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0055000000000000);
#9866=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0055000000000000);
#9867=PRECISION_QUALIFIER(4);
#9873=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2015000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9865
, #9866, #9867))REPRESENTATION_ITEM('minor diameter'));
#9874=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9845, #9846, #9847, #9848, #9849,
#9850, #9859, #9864, #9873), #29);
#9875=PROPERTY_DEFINITION_REPRESENTATION(#9841, #9874);
#9876=SHAPE_DEFINING_RELATIONSHIP('', 'applied shape', #3569, #9887);
#9877=APPLIED_AREA('', $, #38, .T.);
#9878=PROPERTY_DEFINITION('', $, #9877);
#9883=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999), #10)REPRESENTATION_ITEM('length'));
#9884=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9845, #9883), #29);
#9885=PROPERTY_DEFINITION_REPRESENTATION(#9878, #9884);
#9886=PRODUCT_DEFINITION_SHAPE('', $, #9840);
#9887=SHAPE_ASPECT('', 'partial area occurrence', #9886, .T.);
#9888=SHAPE_DEFINING_RELATIONSHIP('', 'applied area usage', #9877, #9887);

```

```

#9894=(CHARACTERIZED_OBJECT('', 'thread')FEATURE_DEFINITION() INSTANCED_FEATURE
() SHAPE_ASPECT('', 'thread', #38, .T.)THREAD());
#9895=PROPERTY_DEFINITION('', $, #9894);
#9896=CARTESIAN_POINT('', (130.0, 0.0, 60.0));
#9897=DIRECTION('', (0.0, -1.0, 0.0));
#9898=DIRECTION('', (1.0, 0.0, 0.0));
#9899=AXIS2_PLACEMENT_3D('orientation', #9896, #9897, #9898);
#9900=DESCRIPTIVE_REPRESENTATION_ITEM('thread side', 'internal');
#9901=DESCRIPTIVE_REPRESENTATION_ITEM('form', 'UNC');
#9902=DESCRIPTIVE_REPRESENTATION_ITEM('fit class', '2B');
#9903=DESCRIPTIVE_REPRESENTATION_ITEM('hand', 'right');
#9904=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0), #11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9905=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0024000000000000);
#9906=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0024000000000000);
#9907=PRECISION_QUALIFIER(4);
#9913=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2199000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9905
, #9906, #9907))REPRESENTATION_ITEM('pitch diameter'));
#9918=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000), #10)REPRESENTATION_ITEM('major
diameter'));
#9919=STANDARD_UNCERTAINTY('upper limit', 'plus', 0.0055000000000000);
#9920=STANDARD_UNCERTAINTY('lower limit', 'minus', 0.0055000000000000);
#9921=PRECISION_QUALIFIER(4);
#9927=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2015000000000000), #10)QUALIFIED_REPRESENTATION_ITEM((#9919
, #9920, #9921))REPRESENTATION_ITEM('minor diameter'));
#9928=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9899, #9900, #9901, #9902, #9903,
#9904, #9913, #9918, #9927), #29);
#9929=PROPERTY_DEFINITION_REPRESENTATION(#9895, #9928);
#9930=SHAPE_DEFINING_RELATIONSHIP('', 'applied shape', #3709, #9941);
#9931=APPLIED_AREA('', $, #38, .T.);
#9932=PROPERTY_DEFINITION('', $, #9931);
#9937=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999), #10)REPRESENTATION_ITEM('length'));
#9938=SHAPE_REPRESENTATION_WITH_PARAMETERS('', (#9899, #9937), #29);
#9939=PROPERTY_DEFINITION_REPRESENTATION(#9932, #9938);
#9940=PRODUCT_DEFINITION_SHAPE('', $, #9894);
#9941=SHAPE_ASPECT('', 'partial area occurrence', #9940, .T.);
#9942=SHAPE_DEFINING_RELATIONSHIP('', 'applied area usage', #9931, #9941);
#9948=(CHARACTERIZED_OBJECT('', 'thread')FEATURE_DEFINITION() INSTANCED_FEATURE
() SHAPE_ASPECT('', 'thread', #38, .T.)THREAD());
#9949=PROPERTY_DEFINITION('', $, #9948);
#9950=CARTESIAN_POINT('', (130.0, 0.0, 120.0));
#9951=DIRECTION('', (0.0, -1.0, 0.0));
#9952=DIRECTION('', (1.0, 0.0, 0.0));
#9953=AXIS2_PLACEMENT_3D('orientation', #9950, #9951, #9952);
#9954=DESCRIPTIVE_REPRESENTATION_ITEM('thread side', 'internal');
#9955=DESCRIPTIVE_REPRESENTATION_ITEM('form', 'UNC');
#9956=DESCRIPTIVE_REPRESENTATION_ITEM('fit class', '2B');
#9957=DESCRIPTIVE_REPRESENTATION_ITEM('hand', 'right');

```

```

#9958=(MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UNIT(RATIO_MEASURE(20.0),#11
)
RATIO_MEASURE_WITH_UNIT()REPRESENTATION_ITEM('number of threads'));
#9959=STANDARD_UNCERTAINTY('upper limit','plus',0.0024000000000000);
#9960=STANDARD_UNCERTAINTY('lower limit','minus',0.0024000000000000);
#9961=PRECISION_QUALIFIER(4);
#9967=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2199000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9959
,#9960,#9961))REPRESENTATION_ITEM('pitch diameter'));
#9972=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2500000000000000),#10)REPRESENTATION_ITEM('major
diameter'));
#9973=STANDARD_UNCERTAINTY('upper limit','plus',0.0055000000000000);
#9974=STANDARD_UNCERTAINTY('lower limit','minus',0.0055000000000000);
#9975=PRECISION_QUALIFIER(4);
#9981=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(0.2015000000000000),#10)QUALIFIED_REPRESENTATION_ITEM((#9973
,#9974,#9975))REPRESENTATION_ITEM('minor diameter'));
#9982=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9953,#9954,#9955,#9956,#9957,
#9958,#9967,#9972,#9981),#29);
#9983=PROPERTY_DEFINITION_REPRESENTATION(#9949,#9982);
#9984=SHAPE_DEFINING_RELATIONSHIP('','applied shape',#3639,#9988);
#9985=APPLIED_AREA('','$,#38,.T.);
#9986=PROPERTY_DEFINITION('','$,#9985);
#9988=SHAPE_ASPECT('','',#9994,.T.);
#9991=(LENGTH_MEASURE_WITH_UNIT()MEASURE_REPRESENTATION_ITEM()MEASURE_WITH_UN
IT(LENGTH_MEASURE(12.699999999999999),#10)REPRESENTATION_ITEM('length'));
#9992=SHAPE_REPRESENTATION_WITH_PARAMETERS('',(#9953,#9991),#29);
#9993=PROPERTY_DEFINITION_REPRESENTATION(#9986,#9992);
#9994=PRODUCT_DEFINITION_SHAPE('','$,#9948);
#9995=SHAPE_ASPECT('','partial area occurrence',#9994,.T.);
#9996=SHAPE_DEFINING_RELATIONSHIP('','applied area usage',#9985,#9995);
#9997=DATUM_FEATURE('','$,#38,.T.);
#9998=PROPERTY_DEFINITION('','$,#9997);
#9999=REPRESENTATION('',(1990),#29);
#10000=PROPERTY_DEFINITION_REPRESENTATION(#9998,#9999);
#10001=DATUM('B','$,#38,.F.,'B');
#10002=SHAPE_ASPECT_RELATIONSHIP('','$,#9997,#10001);
#10003=REFERENCED_MODIFIED_DATUM(2,#10001,.MAXIMUM_MATERIAL_CONDITION.);
#10004=REFERENCED_MODIFIED_DATUM(3,#5534,.MAXIMUM_MATERIAL_CONDITION.);
#10005=PRECISION_QUALIFIER(3);
#10006=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#10007=MEASURE_QUALIFICATION('','',#10006,(#10005));
#10012=(GEOMETRIC_TOLERANCE('position','',#10006,#7862)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10013=REFERENCED_MODIFIED_DATUM(2,#5534,.MAXIMUM_MATERIAL_CONDITION.);
#10014=DATUM_REFERENCE(3,#5555);
#10015=PRECISION_QUALIFIER(3);
#10016=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10017=MEASURE_QUALIFICATION('','',#10016,(#10015));

```

ISO 10303-223:2008 (E)

```

#10021=(GEOMETRIC_TOLERANCE('surface
profile', '', #10016, #3289)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543, #100
13, #10014))SURFACE_PROFILE_TOLERANCE());
#10022=PRECISION_QUALIFIER(3);
#10023=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.5000000000000000), #10);
#10024=MEASURE_QUALIFICATION('', '', #10023, (#10022));
#10025=CONCENTRICITY_TOLERANCE('concentricity', '', #10023, #2405, (#5536));
#10026=TOLERANCE_ZONE_FORM('diametral');
#10027=TOLERANCE_ZONE('', $, #38, .F., (#10025), #10026);
#10028=PRECISION_QUALIFIER(3);
#10029=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.5000000000000000), #10);
#10030=MEASURE_QUALIFICATION('', '', #10029, (#10028));
#10035=(GEOMETRIC_TOLERANCE('position', '', #10029, #6540)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10036=TOLERANCE_ZONE_FORM('diametral');
#10037=TOLERANCE_ZONE('', $, #38, .F., (#10035), #10036);
#10038=PRECISION_QUALIFIER(3);
#10039=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10040=MEASURE_QUALIFICATION('', '', #10039, (#10038));
#10045=(GEOMETRIC_TOLERANCE('position', '', #10039, #2469)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10046=TOLERANCE_ZONE_FORM('diametral');
#10047=TOLERANCE_ZONE('', $, #38, .F., (#10045), #10046);
#10048=PRECISION_QUALIFIER(3);
#10049=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10050=MEASURE_QUALIFICATION('', '', #10049, (#10048));
#10055=(GEOMETRIC_TOLERANCE('position', '', #10049, #2533)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10056=TOLERANCE_ZONE_FORM('diametral');
#10057=TOLERANCE_ZONE('', $, #38, .F., (#10055), #10056);
#10058=PRECISION_QUALIFIER(3);
#10059=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10060=MEASURE_QUALIFICATION('', '', #10059, (#10058));
#10065=(GEOMETRIC_TOLERANCE('position', '', #10059, #2599)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10066=TOLERANCE_ZONE_FORM('diametral');
#10067=TOLERANCE_ZONE('', $, #38, .F., (#10065), #10066);
#10068=PRECISION_QUALIFIER(3);
#10069=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10070=MEASURE_QUALIFICATION('', '', #10069, (#10068));
#10075=(GEOMETRIC_TOLERANCE('position', '', #10069, #2663)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10076=TOLERANCE_ZONE_FORM('diametral');
#10077=TOLERANCE_ZONE('', $, #38, .F., (#10075), #10076);
#10078=PRECISION_QUALIFIER(3);
#10079=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10080=MEASURE_QUALIFICATION('', '', #10079, (#10078));

```

```

#10085=(GEOMETRIC_TOLERANCE('position','',#10079,#2859)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10086=TOLERANCE_ZONE_FORM('diametral');
#10087=TOLERANCE_ZONE('','$,#38,.F.,(#10085),#10086);
#10088=PRECISION_QUALIFIER(3);
#10089=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#10090=MEASURE_QUALIFICATION('','',#10089,(#10088));
#10095=(GEOMETRIC_TOLERANCE('position','',#10089,#2859)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10096=TOLERANCE_ZONE_FORM('diametral');
#10097=TOLERANCE_ZONE('','$,#38,.F.,(#10095),#10096);
#10098=PRECISION_QUALIFIER(3);
#10099=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10100=MEASURE_QUALIFICATION('','',#10099,(#10098));
#10105=(GEOMETRIC_TOLERANCE('position','',#10099,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10106=TOLERANCE_ZONE_FORM('diametral');
#10107=TOLERANCE_ZONE('','$,#38,.F.,(#10105),#10106);
#10108=PRECISION_QUALIFIER(3);
#10109=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10110=MEASURE_QUALIFICATION('','',#10109,(#10108));
#10115=(GEOMETRIC_TOLERANCE('position','',#10109,#3119)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10116=TOLERANCE_ZONE_FORM('diametral');
#10117=TOLERANCE_ZONE('','$,#38,.F.,(#10115),#10116);
#10118=PRECISION_QUALIFIER(3);
#10119=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#10120=MEASURE_QUALIFICATION('','',#10119,(#10118));
#10125=(GEOMETRIC_TOLERANCE('position','',#10119,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10126=TOLERANCE_ZONE_FORM('diametral');
#10127=TOLERANCE_ZONE('','$,#38,.F.,(#10125),#10126);
#10128=PRECISION_QUALIFIER(3);
#10129=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#10130=MEASURE_QUALIFICATION('','',#10129,(#10128));
#10135=(GEOMETRIC_TOLERANCE('position','',#10129,#2989)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10136=TOLERANCE_ZONE_FORM('diametral');
#10137=TOLERANCE_ZONE('','$,#38,.F.,(#10135),#10136);
#10138=DATUM_REFERENCE(3,#5534);
#10139=PRECISION_QUALIFIER(3);
#10140=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10141=MEASURE_QUALIFICATION('','',#10140,(#10139));
#10146=(GEOMETRIC_TOLERANCE('position','',#10140,#3569)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10138)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10147=TOLERANCE_ZONE_FORM('diametral');

```

ISO 10303-223:2008 (E)

```
#10148=TOLERANCE_ZONE('', $, #38, .F., (#10146), #10147);
#10149=PRECISION_QUALIFIER(3);
#10150=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10151=MEASURE_QUALIFICATION('', '#10150', (#10149));
#10156=(GEOMETRIC_TOLERANCE('position', '#10150', #3709) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003, #10138)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10157=TOLERANCE_ZONE_FORM('diametral');
#10158=TOLERANCE_ZONE('', $, #38, .F., (#10156), #10157);
#10159=PRECISION_QUALIFIER(3);
#10160=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10161=MEASURE_QUALIFICATION('', '#10160', (#10159));
#10166=(GEOMETRIC_TOLERANCE('position', '#10160', #3499) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003, #10138)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10167=TOLERANCE_ZONE_FORM('diametral');
#10168=TOLERANCE_ZONE('', $, #38, .F., (#10166), #10167);
#10169=PRECISION_QUALIFIER(3);
#10170=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10171=MEASURE_QUALIFICATION('', '#10170', (#10169));
#10176=(GEOMETRIC_TOLERANCE('position', '#10170', #3639) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003, #10138)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10177=TOLERANCE_ZONE_FORM('diametral');
#10178=TOLERANCE_ZONE('', $, #38, .F., (#10176), #10177);
#10179=PRECISION_QUALIFIER(3);
#10180=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10181=MEASURE_QUALIFICATION('', '#10180', (#10179));
#10186=(GEOMETRIC_TOLERANCE('position', '#10180', #3569) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10187=TOLERANCE_ZONE_FORM('diametral');
#10188=TOLERANCE_ZONE('', $, #38, .F., (#10186), #10187);
#10189=PRECISION_QUALIFIER(3);
#10190=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10191=MEASURE_QUALIFICATION('', '#10190', (#10189));
#10196=(GEOMETRIC_TOLERANCE('position', '#10190', #3709) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10197=TOLERANCE_ZONE_FORM('diametral');
#10198=TOLERANCE_ZONE('', $, #38, .F., (#10196), #10197);
#10199=PRECISION_QUALIFIER(3);
#10200=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10201=MEASURE_QUALIFICATION('', '#10200', (#10199));
#10206=(GEOMETRIC_TOLERANCE('position', '#10200', #3499) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #10003)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10207=TOLERANCE_ZONE_FORM('diametral');
#10208=TOLERANCE_ZONE('', $, #38, .F., (#10206), #10207);
#10209=PRECISION_QUALIFIER(3);
#10210=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000), #10);
#10211=MEASURE_QUALIFICATION('', '#10210', (#10209));
```



```

#10216=(GEOMETRIC_TOLERANCE('position','',#10210,#3639)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10217=TOLERANCE_ZONE_FORM('diametral');
#10218=TOLERANCE_ZONE('','$,#38,.F.,(#10216),#10217);
#10219=PRECISION_QUALIFIER(3);
#10220=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10221=MEASURE_QUALIFICATION('','',#10220,(#10219));
#10226=(GEOMETRIC_TOLERANCE('position','',#10220,#2729)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003,#10004)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10227=TOLERANCE_ZONE_FORM('diametral');
#10228=TOLERANCE_ZONE('','$,#38,.F.,(#10226),#10227);
#10229=PRECISION_QUALIFIER(3);
#10230=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.2500000000000000),#10);
#10231=MEASURE_QUALIFICATION('','',#10230,(#10229));
#10236=(GEOMETRIC_TOLERANCE('position','',#10230,#2729)GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATER
IAL_CONDITION.)POSITION_TOLERANCE());
#10237=TOLERANCE_ZONE_FORM('diametral');
#10238=TOLERANCE_ZONE('','$,#38,.F.,(#10236),#10237);
#10239=PRECISION_QUALIFIER(3);
#10240=MEASURE_WITH_UNIT(LENGTH_MEASURE(12.0),#10);
#10241=MEASURE_QUALIFICATION('','',#10240,(#10239));
#10242=PARALLELISM_TOLERANCE('parallelism','',#10240,#1361,(#5543));
#10243=PRECISION_QUALIFIER(3);
#10244=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.1000000000000000),#10);
#10245=MEASURE_QUALIFICATION('','',#10244,(#10243));
#10246=PERPENDICULARITY_TOLERANCE('perpendicularity','',#10244,#6113,(#5529))
;
#10247=PRECISION_QUALIFIER(3);
#10248=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.5000000000000000),#10);
#10249=MEASURE_QUALIFICATION('','',#10248,(#10247));
#10250=POSITION_TOLERANCE('position','',#10248,#1957,(#5543,#10013,#10014));
#10251=TOLERANCE_ZONE_FORM('diametral');
#10252=TOLERANCE_ZONE('','$,#38,.F.,(#10250),#10251);
#10253=PRECISION_QUALIFIER(3);
#10254=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.5000000000000000),#10);
#10255=MEASURE_QUALIFICATION('','',#10254,(#10253));
#10256=POSITION_TOLERANCE('position','',#10254,#1917,(#5543,#10013,#10014));
#10257=TOLERANCE_ZONE_FORM('diametral');
#10258=TOLERANCE_ZONE('','$,#38,.F.,(#10256),#10257);
#10259=PRECISION_QUALIFIER(3);
#10260=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000),#10);
#10261=MEASURE_QUALIFICATION('','',#10260,(#10259));
#10262=GEOMETRIC_TOLERANCE('flatness','',#10260,#335);
#10263=PRECISION_QUALIFIER(3);
#10264=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000),#10);
#10265=MEASURE_QUALIFICATION('','',#10264,(#10263));
#10270=(GEOMETRIC_TOLERANCE('position','',#10264,#143)GEOMETRIC_TOLERANCE_WIT
H_DATUM_REFERENCE((#5529,#10003)MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERI
AL_CONDITION.)POSITION_TOLERANCE());
#10271=PRECISION_QUALIFIER(3);

```

ISO 10303-223:2008 (E)

```
#10272=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000),#10);
#10273=MEASURE_QUALIFICATION('','',#10272,(#10271));
#10274=GEOMETRIC_TOLERANCE('flatness','',#10272,#1110);
#10275=PRECISION_QUALIFIER(3);
#10276=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.3000000000000000),#10);
#10277=MEASURE_QUALIFICATION('','',#10276,(#10275));
#10282=(GEOMETRIC_TOLERANCE('position','',#10276,#6214)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#10003,#10004))MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMUM_MATERIAL_CONDITION.)POSITION_TOLERANCE());
#10283=PRECISION_QUALIFIER(3);
#10284=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10285=MEASURE_QUALIFICATION('','',#10284,(#10283));
#10289=(GEOMETRIC_TOLERANCE('surface profile','',#10284,#3789)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#10013,#10014))SURFACE_PROFILE_TOLERANCE());
#10290=PRECISION_QUALIFIER(3);
#10291=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10292=MEASURE_QUALIFICATION('','',#10291,(#10290));
#10296=(GEOMETRIC_TOLERANCE('surface profile','',#10291,#3805)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#10013,#10014))SURFACE_PROFILE_TOLERANCE());
#10297=PRECISION_QUALIFIER(3);
#10298=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10299=MEASURE_QUALIFICATION('','',#10298,(#10297));
#10303=(GEOMETRIC_TOLERANCE('surface profile','',#10298,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#10003,#10004))SURFACE_PROFILE_TOLERANCE());
#10304=PRECISION_QUALIFIER(3);
#10305=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10306=MEASURE_QUALIFICATION('','',#10305,(#10304));
#10310=(GEOMETRIC_TOLERANCE('surface profile','',#10305,#2264)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#10003,#10004))SURFACE_PROFILE_TOLERANCE());
#10311=PRECISION_QUALIFIER(3);
#10312=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000),#10);
#10313=MEASURE_QUALIFICATION('','',#10312,(#10311));
#10317=(GEOMETRIC_TOLERANCE('surface profile','',#10312,#4309)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543,#10013,#10014))SURFACE_PROFILE_TOLERANCE());
#10318=PRECISION_QUALIFIER(3);
#10319=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10320=MEASURE_QUALIFICATION('','',#10319,(#10318));
#10324=(GEOMETRIC_TOLERANCE('surface profile','',#10319,#2221)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#10003,#10004))SURFACE_PROFILE_TOLERANCE());
#10325=PRECISION_QUALIFIER(3);
#10326=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
#10327=MEASURE_QUALIFICATION('','',#10326,(#10325));
#10331=(GEOMETRIC_TOLERANCE('surface profile','',#10326,#2221)GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529,#10003,#10004))SURFACE_PROFILE_TOLERANCE());
#10332=PRECISION_QUALIFIER(3);
#10333=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0),#10);
```

```

#10334=MEASURE_QUALIFICATION('', '#10333', (#10332));
#10338=(GEOMETRIC_TOLERANCE('surface
profile', '#10333', #2264) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #100
03, #10004)) SURFACE_PROFILE_TOLERANCE());
#10339=PRECISION_QUALIFIER(3);
#10340=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.0), #10);
#10341=MEASURE_QUALIFICATION('', '#10340', (#10339));
#10345=(GEOMETRIC_TOLERANCE('surface
profile', '#10340', #2264) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5529, #100
03, #10004)) SURFACE_PROFILE_TOLERANCE());
#10346=PRECISION_QUALIFIER(3);
#10347=MEASURE_WITH_UNIT(LENGTH_MEASURE(1.5000000000000000), #10);
#10348=MEASURE_QUALIFICATION('', '#10347', (#10346));
#10353=(GEOMETRIC_TOLERANCE('position', '#10347', #7862) GEOMETRIC_TOLERANCE_WI
TH_DATUM_REFERENCE((#5543, #10003, #10004)) MODIFIED_GEOMETRIC_TOLERANCE(.MAXIMU
M_MATERIAL_CONDITION.) POSITION_TOLERANCE());
#10354=PRECISION_QUALIFIER(3);
#10355=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10356=MEASURE_QUALIFICATION('', '#10355', (#10354));
#10360=(GEOMETRIC_TOLERANCE('surface
profile', '#10355', #4643) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543, #100
13, #10014)) SURFACE_PROFILE_TOLERANCE());
#10361=PRECISION_QUALIFIER(3);
#10362=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10363=MEASURE_QUALIFICATION('', '#10362', (#10361));
#10367=(GEOMETRIC_TOLERANCE('surface
profile', '#10362', #4783) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543, #100
13, #10014)) SURFACE_PROFILE_TOLERANCE());
#10368=PRECISION_QUALIFIER(3);
#10369=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.0100000000000000), #10);
#10370=MEASURE_QUALIFICATION('', '#10369', (#10368));
#10371=GEOMETRIC_TOLERANCE('flatness', '#10369', #5527);
#10372=PRECISION_QUALIFIER(3);
#10373=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10374=MEASURE_QUALIFICATION('', '#10373', (#10372));
#10378=(GEOMETRIC_TOLERANCE('surface
profile', '#10373', #4309) GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE((#5543, #100
13, #10014)) SURFACE_PROFILE_TOLERANCE());
#10379=PRECISION_QUALIFIER(3);
#10380=MEASURE_WITH_UNIT(LENGTH_MEASURE(0.8000000000000000), #10);
#10381=MEASURE_QUALIFICATION('', '#10380', (#10379));
#10382=SYMMETRY_TOLERANCE('symmetry', '#10380', #9080, (#5550));

#10408=SECURITY_CLASSIFICATION_LEVEL('unclassified');
#10409=SECURITY_CLASSIFICATION('classification', 'classify product
version', #10408);
#10410=APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT(#10409, (#35));

#10522=MEASURE_WITH_UNIT(COUNT_MEASURE(1.0), #30);
#10523=PRODUCT_DEFINITION('3/8 OD X 4 5/16 LG', '4130 STEEL COND D & 4 PER
MIL-S-6758', #35, #36);
#10524=MATERIAL_DESIGNATION('4130 STEEL COND D & 4 PER MIL-S-6758', (#10523));

```

## ISO 10303-223:2008 (E)

```
#10525=MAKE_FROM_USAGE_OPTION('1','','product material',#37,#10523,1,'primary  
material',#10522);
```

```
#10528=VERSIONED_ACTION_REQUEST('','','','');  
#10529=ACTION_DIRECTIVE('','','','',( #10528));  
#10530=ACTION_METHOD('','','','');  
#10531=DIRECTED_ACTION('project order','order for project',#10530,#10529);
```

```
#10540=APPLIED_ACTION_ASSIGNMENT(#10531,(#35));  
#10550=APPLIED_APPROVAL_ASSIGNMENT(#15020,(#10531));
```

```
#15020=APPROVAL(#15021,'');  
#15021=APPROVAL_STATUS('approved');  
#15022=APPROVAL_DATE_TIME(#15023,#15020);  
#15023=DATE(2006);
```

```
ENDSEC;  
END-ISO-10303-21;
```

## Bibliography

- [1] *IDEF0 Federal Information Processing Standards Publication 183, Integration Definition for Functional Modeling (IDEF0)*, FIPS PUB 183, National Institute of Standards and Technology, December 1993.
- [2] *Principles of Purchasing Castings For Buyers, Specifiers, Designers, Engineers, Producers and Users of Castings*, American Foundry Society, 2002.
- [3] ISO 639-2, *Codes for the representation of names of languages — Part 2: Alpha-3 code*
- [4] ISO/IEC 646, *Information technology — ISO 7-bit coded character set for information interchange*
- [5] ISO 3098-2, *Technical product documentation — Lettering — Part 2: Latin alphabet, numerals and marks*
- [6] ISO 3166-1, *Codes for the representation of names of countries and their subdivisions — Part 1: Country codes*
- [7] ISO/IEC 6937, *Information technology — Coded graphic character set for text communication — Latin alphabet*
- [8] ISO 6983 (all parts), *Industrial automation systems and integration — Numerical control of machines — Program format and definitions of address words*
- [9] ISO/IEC 8859-1, *Information technology — 8-bit single-byte coded graphic character sets — Part 1; Latin alphabet No. 1*
- [10] ISO 10303-44, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resource: Product structure configuration*
- [11] ISO 10303-202, *Industrial automation systems and integration — Product data representation and exchange — Part 202: Application protocol: Associative draughting*
- [12] ISO 10303-203, *Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol. Configuration controlled 3D design of mechanical parts and assemblies (modular version)*
- [13] ISO 10303-214, *Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol. Core data for automotive mechanical design processes*
- [14] ISO 10303-219, *Industrial automation systems and integration — Product data representation and exchange — Part 219: Application protocol: Dimensional inspection information exchange*

ISO 10303-223:2008 (E)

- [15] ISO/IEC 10646, *Information technology — Universal Multiple-Octet Coded Character Set (UCS)*
- [16] ISO 13399 (all parts), *Cutting tool data representation and exchange*
- [17] ISO 13584 (all parts), *Industrial automation systems and integration — Parts library*
- [18] IEC 61286, *Information technology — Coded graphic character set for use in the preparation of documents used in electrotechnology and for information interchange*

## Index

AAM .....	9
action	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1413
action_assignment	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1413
action_directive	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1413
mapping specification .....	607
action_item	
AIM diagram .....	1927
AIM EXPRESS long listing type .....	1397
AIM EXPRESS short listing type .....	1005
action_method	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1414
action_method_assignment	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1414
action_method_item	
AIM diagram .....	1929
AIM EXPRESS long listing type .....	1397
AIM EXPRESS short listing type .....	1005
action_method_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1414
action_method_role	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1414
action_method_with_associated_documents	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1414
action_property	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1414
action_property_relationship	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1414
action_property_representation	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1415
action_relationship	
AIM diagram .....	1927

AIM EXPRESS long listing entity .....	1415
AIM EXPRESS short listing imported entity modifications .....	1325
action_request_assignment	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1415
action_request_item	
AIM diagram .....	1931
AIM EXPRESS long listing type .....	1397
AIM EXPRESS short listing type .....	1006
action_request_solution	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1415
action_resource	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1416
action_resource_relationship	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1416
action_resource_requirement	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1416
action_resource_type	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1416
action_status	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1416
Activit_result_record	
application object.....	33
Activity	
application assertion .....	295, 296, 297, 310, 350, 371
application object.....	30
ARM diagram .....	1881
mapping specification.....	863
activity_execution_result	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1416
AIM EXPRESS short listing entity .....	1011
mapping specification.....	925
Activity_execution_result	
application assertion .....	295, 297, 349
application object.....	32
ARM diagram .....	1883
mapping specification.....	925
activity_result_record	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1417
AIM EXPRESS short listing entity .....	1012



mapping specification.....	926
Activity_result_record	
application assertion .....	297
ARM diagram .....	1883
mapping specification.....	926
acyclic_mapped_representation	
AIM EXPRESS long listing FUNCTION .....	1749
acyclic_product_definition_relationship	
AIM EXPRESS long listing FUNCTION .....	1750
address	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1417
mapping specification.....	842
Address	
application assertion .....	341, 347
application object.....	33
ARM diagram .....	1854
mapping specification.....	842
advanced_B_rep	
ARM diagram .....	1861
advanced_brep_shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1417
advanced_face	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1418
ahead_or_behind	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1398
aid_type	
ARM diagram .....	1852
AIM.....	9
allowed_time	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1420
AIM EXPRESS short listing entity .....	1012
mapping specification.....	868, 905
Allowed_time	
application object.....	34
ARM diagram .....	1881
mapping specification.....	868
alternate_action_method_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1420
AIM EXPRESS short listing entity .....	1014
Alternate_activity	
application assertion .....	297
application object.....	35
© ISO 2008 — All rights reserved	2241

ARM diagram .....	1881
mapping specification .....	870
alternate_plan_relationship	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1420
AIM EXPRESS short listing entity .....	1014
Alternate_process_plan	
application assertion .....	297
application object .....	36
ARM diagram .....	1880
mapping specification .....	914
amount_of_substance_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1398
ancillary_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1421
AIM EXPRESS short listing entity .....	1015
mapping specification .....	870, 871
Ancillary_activity	
application object .....	36
ARM diagram .....	1884
mapping specification .....	870
ancillary_setup	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1421
AIM EXPRESS short listing entity .....	1015
mapping specification .....	906
Ancillary_setup	
application object .....	36
ARM diagram .....	1881
mapping specification .....	871
angle_relator	
AIM diagram .....	1926
AIM EXPRESS long listing type .....	1398
Angular_dimension_tolerance	
application object .....	37
ARM diagram .....	1877
mapping specification .....	777
angular_location	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1421
mapping specification .....	777, 778
angular_size	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1421
mapping specification .....	778
Angular_size_dimension_tolerance	

application object.....	38
ARM diagram.....	1877
mapping specification.....	778
angularity_tolerance	
AIM diagram.....	1924
AIM EXPRESS long listing entity.....	1421
mapping specification.....	788
AP.....	9
apex	
AIM diagram.....	1925
AIM EXPRESS long listing entity.....	1421
Application.....	5, 6
Application activity model.....	5
Application interpreted model.....	5
Application protocol.....	5
Application reference model.....	5, 6
application_context	
AIM EXPRESS long listing entity.....	1421
AIM EXPRESS short listing imported entity modifications.....	1325
application_context_element	
AIM EXPRESS long listing entity.....	1422
application_context_requires_ap_definition	
AIM EXPRESS long listing rule.....	1737
AIM EXPRESS short listing rules.....	1329
application_protocol_definition	
AIM EXPRESS long listing entity.....	1422
AIM EXPRESS short listing imported entity modifications.....	1325
applied_action_assignment	
AIM diagram.....	1927
AIM EXPRESS long listing entity.....	1422
AIM EXPRESS short listing entity.....	1016
applied_action_method_assignment	
AIM diagram.....	1929
AIM EXPRESS long listing entity.....	1422
AIM EXPRESS short listing entity.....	1016
applied_action_request_assignment	
AIM diagram.....	1931
AIM EXPRESS long listing entity.....	1422
AIM EXPRESS short listing entity.....	1016
applied_approval_assignment	
AIM diagram.....	1937
AIM EXPRESS long listing entity.....	1423
AIM EXPRESS short listing entity.....	1017
applied_area	
AIM diagram.....	1911
AIM EXPRESS long listing entity.....	1423
mapping specification.....	818
applied_classification_assignment	
© ISO 2008 — All rights reserved	2243

AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1424
AIM EXPRESS short listing entity .....	1017
mapping specification .....	701
applied_date_assignment	
AIM EXPRESS short listing entity .....	1018
applied_date_and_time_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1424
AIM EXPRESS short listing entity .....	1018
applied_date_assignment	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1424
applied_document_assignment	
AIM EXPRESS short listing entity .....	1018
applied_document_reference	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1424
mapping specification .....	756, 769
applied_document_usage_constraint_assignment	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1019
applied_effectivity_assignment	
AIM diagram .....	1906
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1019
applied_external_identification_assignment	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1019
mapping specification .....	764
applied_group_assignment	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1020
applied_identification_assignment	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1020
applied_library_assignment	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1425
AIM EXPRESS short listing entity .....	1021
applied_organization_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1426
AIM EXPRESS short listing entity .....	1022

applied_person_and_organization_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1426
AIM EXPRESS short listing entity .....	1022
applied_security_assignment	
AIM EXPRESS short listing entity .....	1023
applied_security_classification_assignment	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1426
applied_time_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1426
AIM EXPRESS short listing entity .....	1023
approval	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1426
mapping specification .....	843, 846
Approval	
application assertion .....	297, 320, 325, 343, 359
application object .....	39
ARM diagram .....	1854
mapping specification .....	843
approval_assignment	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1427
approval_date_time	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1427
approval_person_organization	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1427
approval_role	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1427
approval_status	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1427
AIM EXPRESS short listing imported entity modifications .....	1326
approved_item	
AIM diagram .....	1937
AIM EXPRESS long listing type .....	1398
AIM EXPRESS short listing type .....	1006
area_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1398
ARM .....	6, 9
assembly	
AIM diagram .....	1928
© ISO 2008 — All rights reserved	2245

AIM EXPRESS long listing entity .....	1427
AIM EXPRESS short listing entity .....	1023
mapping specification .....	871
Assembly	
application assertion .....	328
application object .....	40
ARM diagram .....	1886
mapping specification .....	871
assembly_component_usage	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1428
associated_surface	
AIM EXPRESS long listing FUNCTION .....	1751
attribute_type	
AIM diagram .....	1945
AIM EXPRESS long listing type .....	1398
axis1_placement	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1428
axis2_placement	
AIM diagram .....	1916
AIM EXPRESS long listing type .....	1398
axis2_placement_2d	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1428
axis2_placement_3d	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1429
b_spline_curve	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1429
b_spline_curve_form	
AIM diagram .....	1919
AIM EXPRESS long listing type .....	1399
b_spline_curve_with_knots	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1429
b_spline_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1430
b_spline_surface_form	
AIM diagram .....	1921
AIM EXPRESS long listing type .....	1399
b_spline_surface_with_knots	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1430
bag_to_set	
AIM EXPRESS long listing FUNCTION .....	1751

bankout_frame	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1431
AIM EXPRESS short listing entity .....	1024
mapping specification .....	706
Bankout_frame	
application object .....	40
ARM diagram .....	1895
mapping specification .....	706
base_axis	
AIM EXPRESS long listing FUNCTION .....	1751
Base_shape	
application assertion .....	342
application object .....	41
ARM diagram .....	1859
mapping table .....	957
bench_inspection	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1431
AIM EXPRESS short listing entity .....	1025
mapping specification .....	873
Bench_inspection	
application object .....	41
ARM diagram .....	1889
mapping specification .....	873
bezier_curve	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1431
bezier_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1431
boolean_choose	
AIM EXPRESS long listing FUNCTION .....	1752
boolean_operand	
AIM diagram .....	1923
AIM EXPRESS long listing type .....	1399
boss	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1431
boss_top	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1434
boundary_condition	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1435
AIM EXPRESS short listing entity .....	1025
mapping specification .....	973
Boundary_condition	
© ISO 2008 — All rights reserved	2247

ISO 10303-223:2008 (E)

application assertion .....	298, 369
application object.....	41
ARM diagram .....	1892
mapping specification.....	973
bounded_curve	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1436
bounded_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1436
brep_with_voids	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1436
BSU	
application object.....	41
ARM diagram .....	1901
mapping specification.....	698
build_2axes	
AIM EXPRESS long listing FUNCTION .....	1753
build_axes	
AIM EXPRESS long listing FUNCTION .....	1753
calendar_date	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1436
cartesian_point	
AIM diagram .....	1915
AIM EXPRESS long listing entity .....	1437
mapping specification.....	960
cartesian_transformation_operator	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1437
cartesian_transformation_operator_3d	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1437
cast_part	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1437
AIM EXPRESS short listing entity .....	1026
mapping specification.....	847
Cast_part	
application assertion .....	295, 298, 339, 342, 371, 377
application object.....	42
ARM diagram .....	1850
mapping specification.....	397, 847
cast_part_with_rigging	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1437
AIM EXPRESS short listing entity .....	1027



Cast_part_with_rigging	
application assertion .....	295, 298, 328, 342, 371
application object.....	42
ARM diagram.....	1850
cast_part_with_rigging_part	
mapping specification.....	848
Cast_part_with_rigging_part	
mapping specification.....	848
casting_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1438
AIM EXPRESS short listing entity .....	1028
mapping specification.....	873
Casting_activity	
application assertion .....	298, 333, 349, 350
application object.....	43
ARM diagram.....	1884
mapping specification.....	873
casting_design_feature	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1438
AIM EXPRESS short listing entity .....	1029
Casting_design_feature	
application assertion .....	298, 299
application object.....	43
ARM diagram.....	1862
mapping specification.....	381
casting_envelope_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1439
AIM EXPRESS short listing entity .....	1030
mapping specification.....	957
Casting_envelope_requirement	
application assertion .....	299
application object.....	44
ARM diagram.....	1859
mapping specification.....	957
casting_equipment	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1440
AIM EXPRESS short listing entity .....	1031
mapping specification.....	718
casting_equipment_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1503
AIM EXPRESS short listing entity .....	1033
mapping specification.....	880
casting_equipment_setup	
© ISO 2008 — All rights reserved	2249

ISO 10303-223:2008 (E)

AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1504
AIM EXPRESS short listing entity .....	1034
mapping specification .....	881
casting_feature_definition	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1440
AIM EXPRESS short listing entity .....	1033
casting_feature_life_cycle	
AIM EXPRESS long listing rule .....	1737
AIM EXPRESS short listing rules .....	1330
casting_feature_size	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1441
AIM EXPRESS short listing entity .....	1035
Casting_features	
units of functionality .....	11
casting_insert	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1442
AIM EXPRESS short listing entity .....	1037
Casting_insert	
application assertion .....	351, 357
application object .....	44
ARM diagram .....	1867
mapping specification .....	383
casting_instanced_feature	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1442
AIM EXPRESS short listing entity .....	1038
mapping table .....	622, 623
casting_method	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1443
AIM EXPRESS short listing entity .....	1038
casting_method_select	
ARM diagram .....	1855
casting_per_order	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1443
AIM EXPRESS short listing entity .....	1039
mapping specification .....	926
Casting_per_order	
application assertion .....	299
application object .....	45
ARM diagram .....	1889
mapping specification .....	926
casting_prerequisites	

units of functionality.....	13
casting_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1443
AIM EXPRESS short listing entity .....	1039, 1042
mapping specification.....	874
casting_process_activity_type	
ARM diagram.....	1884
casting_product_definition	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1445
mapping specification.....	862
casting_service_data	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1446
AIM EXPRESS short listing entity .....	1043
mapping specification.....	560
Casting_service_data	
application assertion .....	300, 301
application object.....	49
ARM diagram.....	1898
mapping specification.....	560
casting_verification	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1447
AIM EXPRESS short listing entity .....	1046
mapping specification.....	927
Casting_verification	
application assertion .....	301, 325
application object.....	51
ARM diagram.....	1889
mapping specification.....	927
Casting_weight_requirement	
application assertion .....	302, 360
application object.....	52
ARM diagram.....	1856
mapping specification.....	729
celsius_temperature_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1399
centre_of_symmetry	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1448
chamfer	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1448
AIM EXPRESS short listing imported entity modifications .....	1326
chamfer_offset	
© ISO 2008 — All rights reserved	2251

AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1449
AIM EXPRESS short listing imported entity modifications .....	1326
chamfer_offset_requires_faces	
AIM EXPRESS long listing rule .....	1737
AIM EXPRESS short listing rules .....	1331
chamfer_requires_faces	
AIM EXPRESS long listing rule .....	1738
AIM EXPRESS short listing rules .....	1331
chaplet	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1451
AIM EXPRESS short listing entity .....	1048
Chaplet	
application assertion .....	302, 366
application object .....	52
ARM diagram .....	1869
mapping specification .....	385
chaplet_pad	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1451
AIM EXPRESS short listing entity .....	1051
Chaplet_pad	
application assertion .....	302, 351, 357
application object .....	54
ARM diagram .....	1869
mapping specification .....	387
characterized_action_definition	
AIM diagram .....	1930
AIM EXPRESS long listing type .....	1399
characterized_definition	
AIM diagram .....	1907
AIM EXPRESS long listing type .....	1400
characterized_material_property	
AIM diagram .....	1908
AIM EXPRESS long listing type .....	1400
characterized_object	
AIM diagram .....	1903
AIM EXPRESS long listing entity .....	1452
AIM EXPRESS short listing imported entity modifications .....	1326
mapping specification .....	819, 820
characterized_product_definition	
AIM diagram .....	1907
AIM EXPRESS long listing type .....	1400
characterized_resource_definition	
AIM diagram .....	1932
AIM EXPRESS long listing type .....	1400
checking_aid_tool	
2252	

AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1452
AIM EXPRESS short listing entity .....	1049
Checking_aid_tool	
application assertion .....	302, 327, 376
application object.....	54
ARM diagram .....	1852
mapping specification.....	929
chemical_element	
ARM diagram .....	1857
chill	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1453
AIM EXPRESS short listing entity .....	1051
Chill	
application assertion .....	302, 358, 367
application object.....	55
ARM diagram .....	1867
mapping specification.....	388
choke	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1453
AIM EXPRESS short listing entity .....	1053
Choke	
application assertion .....	302, 365
application object.....	56
ARM diagram .....	1864
mapping specification.....	390
circle	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1454
circular_closed_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1454
circular_pattern	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1455
circular_runout_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1457
mapping specification.....	788
class	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1457
AIM EXPRESS short listing entity .....	1054
Class_BSU	
application assertion .....	302, 330, 361
application object.....	57
© ISO 2008 — All rights reserved	2253

ISO 10303-223:2008 (E)

ARM diagram .....	1901
mapping specification .....	699
classification_assignment	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1457
classification_item	
AIM diagram .....	1935
AIM EXPRESS long listing type .....	1400
AIM EXPRESS short listing type .....	1006
classification_role	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1457
closed_path_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1457
closed_shell	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1458
closed_shell_reversed	
AIM EXPRESS long listing FUNCTION .....	1753
coaxiality_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1458
common_datum	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1459
mapping specification .....	788
Common_datum	
ARM diagram .....	1879
compatible_dimension	
AIM EXPRESS long listing rule .....	1738
component	
other definitions .....	6
composite_curve	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1459
composite_curve_on_surface	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1459
composite_curve_segment	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1459
composite_hole	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1460
composite_shape_aspect	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1460

composition inspection	
mapping specification .....	934
Composition inspection	
mapping specification .....	934
composition_element	
AIM EXPRESS long listing entity .....	1461
AIM EXPRESS short listing entity .....	1054
Composition_element	
application assertion .....	303, 335
application object .....	58
ARM diagram .....	1857
mapping specification .....	731
composition_element_record	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1461
AIM EXPRESS short listing entity .....	1055
mapping specification .....	933
Composition_element_record	
application assertion .....	303
application object .....	59
ARM diagram .....	1883
mapping specification .....	933
composition_element_requires_name	
AIM EXPRESS long listing rule .....	1738
AIM EXPRESS short listing rules .....	1332
composition_inspection	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1461
AIM EXPRESS short listing entity .....	1056
Composition_inspection	
application assertion .....	303
application object .....	59
ARM diagram .....	1852
Compound_datum	
application assertion .....	303
application object .....	58
mapping specification .....	779
compound_feature	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1462
mapping specification .....	814
Compound_feature	
application assertion .....	303
application object .....	60
ARM diagram .....	1861
mapping specification .....	814
Compound_feature_element	
application assertion .....	303, 304
© ISO 2008 — All rights reserved	2255

application object.....	60
ARM diagram.....	1861
mapping specification.....	818
Compound_feature_relationship	
application assertion.....	304
application object.....	61
ARM diagram.....	1861
mapping specification.....	818
compound_feature_select	
ARM diagram.....	1861
compound_item_definition	
AIM diagram.....	1914
AIM EXPRESS long listing type.....	1400
compound_representation_item	
AIM diagram.....	1914
AIM EXPRESS long listing entity.....	1463
concentricity_tolerance	
AIM diagram.....	1924
AIM EXPRESS long listing entity.....	1463
mapping specification.....	788
condition_or_assumption	
AIM diagram.....	1928
AIM EXPRESS long listing entity.....	1463
AIM EXPRESS short listing entity.....	1057
mapping specification.....	974
Condition_or_assumption	
application assertion.....	304, 368
application object.....	62
ARM diagram.....	1892
mapping specification.....	974
conditional_reverse	
AIM EXPRESS long listing FUNCTION.....	1754
conformance class.....	6
conformance testing.....	6
conic	
AIM diagram.....	1917
AIM EXPRESS long listing entity.....	1464
conical_surface	
AIM diagram.....	1920
AIM EXPRESS long listing entity.....	1464
connected_face_set	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1464
Connection_transition	
application assertion.....	304, 320, 325
application object.....	62
ARM diagram.....	1864
mapping specification.....	392



constraints_composite_curve_on_surface	
AIM EXPRESS long listing FUNCTION .....	1754
constraints_param_b_spline	
AIM EXPRESS long listing FUNCTION .....	1754
context_dependent_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1465
Continuous_process	
application assertion .....	305
application object.....	63
ARM diagram .....	1888
mapping specification.....	915
continuous_process_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1465
AIM EXPRESS short listing entity .....	1058
mapping specification.....	915
continuous_process_type	
ARM diagram .....	1888
controller	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1465
AIM EXPRESS short listing entity .....	1059
mapping specification.....	707
Controller	
application assertion .....	305, 317
application object.....	64
ARM diagram .....	1894
mapping specification.....	707
conversion_based_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1465
cooling_port	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1465
AIM EXPRESS short listing entity .....	1060
mapping specification.....	611
Cooling_port	
application assertion .....	305, 330, 353
application object.....	65
ARM diagram .....	1874
mapping specification.....	611
coordinated_universal_time_offset	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1466
cope_or_drag_type	
ARM diagram .....	1864
core	
© ISO 2008 — All rights reserved	2257

AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1466
AIM EXPRESS short listing entity .....	1061
Core	
application assertion .....	305, 351, 354, 378
application object.....	65
ARM diagram .....	1876
mapping specification.....	393
core_box	
other definitions.....	6
core_cavity	
other definitions.....	6
core_print	
other definitions.....	6
core_assembly	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1467
AIM EXPRESS short listing entity .....	1063
mapping specification.....	932
Core_assembly	
application assertion .....	305, 306, 377
application object.....	67
ARM diagram.....	1887
mapping specification.....	932
core_bench	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1468
AIM EXPRESS short listing entity .....	1064
mapping specification.....	708
Core_bench	
application object.....	68
ARM diagram.....	1895
mapping specification.....	708
core_box_tooling	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1468
AIM EXPRESS short listing entity .....	1065
Core_box_tooling	
application assertion .....	306
application object.....	68
ARM diagram.....	1875
core_box_vent	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1469
AIM EXPRESS short listing entity .....	1066
mapping specification.....	710
Core_box_vent	
application assertion .....	306

application object.....	69
ARM diagram.....	1895
mapping specification.....	710
core_conveyer_system	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1470
AIM EXPRESS short listing entity.....	1069
mapping specification.....	712
Core_conveyer_system	
application assertion.....	306
ARM diagram.....	1895
mapping specification.....	712
core_equipemt_type	
ARM diagram.....	1897
core_equipment	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1472
AIM EXPRESS short listing entity.....	1072
mapping specification.....	714
Core_equipment	
application assertion.....	351
application object.....	71
ARM diagram.....	1895
mapping specification.....	714
core_machine	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1472
AIM EXPRESS short listing entity.....	1072
mapping specification.....	714
Core_machine	
application assertion.....	306
application object.....	71
ARM diagram.....	1895
mapping specification.....	714
core_master	
AIM diagram.....	1905
AIM EXPRESS long listing entity.....	1473
AIM EXPRESS short listing entity.....	1073
Core_master	
application object.....	71
ARM diagram.....	1875
mapping specification.....	398
core_material	
ARM diagram.....	1876
core_print	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1473
AIM EXPRESS short listing entity.....	1074
© ISO 2008 — All rights reserved	2259

Core_print	
application assertion .....	307, 352, 358, 377
application object.....	72
ARM diagram .....	1876
mapping specification.....	400
core_process_type	
ARM diagram .....	1884
core_setter	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1474
AIM EXPRESS short listing entity .....	1076
mapping specification.....	716
Core_setter	
application object.....	73
ARM diagram .....	1895
mapping specification.....	716
core_type	
ARM diagram .....	1876
Coreconveyer_system	
application object.....	70
corrective_action_types	
ARM diagram .....	1890
corrosion_service	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1475
AIM EXPRESS short listing entity .....	1077
corrosion_Service	
mapping specification.....	564
Corrosion_service	
application assertion .....	300, 307
application object.....	73
ARM diagram .....	1898
Corrosion_Service	
mapping specification.....	564
count_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1400
cross_product	
AIM EXPRESS long listing FUNCTION .....	1755
curve	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1476
curve_on_surface	
AIM diagram .....	1917
AIM EXPRESS long listing type .....	1400
Curve_point	
application assertion .....	309
application object.....	74

ARM diagram .....	1858
mapping specification .....	960
curve_weights_positive	
AIM EXPRESS long listing FUNCTION .....	1756
Curved_dimension_tolerance	
application object .....	74
ARM diagram .....	1877
mapping specification .....	780
customer_casting_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1476
AIM EXPRESS short listing entity .....	1078
mapping specification .....	566
Customer_casting_requirement	
application assertion .....	307, 308, 326, 343, 362, 371, 378
application object .....	75
ARM diagram .....	1896
mapping specification .....	566
Customer_customer_tool_design_report	
application assertion .....	309
Customer_design_deficiency_quality_report	
application object .....	77
Customer_design_deficiency_report	
application assertion .....	309
ARM diagram .....	1899
Customer_design_Product_quality_report	
mapping specification .....	753
Customer_order	
application assertion .....	308, 309
application object .....	77
ARM diagram .....	1855
mapping specification .....	823
customer_order_rules	
AIM EXPRESS long listing rule .....	1739
AIM EXPRESS short listing rules .....	1334
customer_requirement_select	
ARM diagram .....	1890
customer_simulation	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1477
mapping specification .....	878
Customer_simulation	
application assertion .....	309
application object .....	80
ARM diagram .....	1891
mapping specification .....	878
Customer_tool_design_report	
application object .....	80
© ISO 2008 — All rights reserved	2261

ARM diagram .....	1899
mapping specification .....	775
cylindrical_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1477
cylindricity_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1477
mapping specification .....	788
data_curve	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1477
AIM EXPRESS short listing entity .....	1081
mapping specification .....	732
Data_curve	
application assertion .....	309, 361
application object .....	80
ARM diagram .....	1858
mapping specification .....	732
data_environment	
AIM diagram .....	1908
AIM EXPRESS long listing entity .....	1478
data_format	
ARM diagram .....	1900
date	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1478
mapping specification .....	844
Date	
application assertion .....	297, 302, 308, 310, 320, 324, 332, 334, 362, 363, 372, 373, 378
application object .....	81
ARM diagram .....	1854
mapping specification .....	844
date_and_time	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1478
date_and_time_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1478
date_and_time_item	
AIM diagram .....	1939
AIM EXPRESS long listing type .....	1401
AIM EXPRESS short listing type .....	1006
date_assignment	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1479
date_item	
AIM diagram .....	1938

AIM EXPRESS long listing type .....	1401
AIM EXPRESS short listing type .....	1007
date_role	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1479
date_time_or_event_occurrence	
AIM diagram .....	1906
AIM EXPRESS long listing type .....	1401
date_time_role	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1479
date_time_select	
AIM diagram .....	1937
AIM EXPRESS long listing type .....	1401
dated_effectivity	
AIM diagram .....	1906
AIM EXPRESS long listing entity .....	1479
mapping specification .....	939, 941
datum	5
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1479
mapping specification .....	780
Datum	
application object .....	81
ARM diagram .....	1879
mapping specification .....	780
datum_established_by_relationships	
mapping specification .....	781
datum_feature	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1480
Datum_feature	
application assertion .....	303, 309
application object .....	82
mapping specification .....	780
Datum_ferature	
ARM diagram .....	1879
datum_reference	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1480
mapping specification .....	780
datum_representation_select	
ARM diagram .....	1879
datum_target	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1480
mapping specification .....	781, 801
Datum_target	
© ISO 2008 — All rights reserved	2263

application assertion .....	309
application object.....	82
ARM diagram .....	1879
mapping specification.....	781
Datum_target_set	
application assertion .....	309
application object.....	83
ARM diagram .....	1879
mapping specification.....	781
day_in_month_number	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1401
defect_prediction	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1480
AIM EXPRESS short listing entity .....	1082
mapping specification.....	976
Defect_prediction	
application assertion .....	310, 370
application object.....	83
ARM diagram .....	1893
mapping specification.....	976
defining_action_method_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1481
AIM EXPRESS short listing entity .....	1083
definitional_representation	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1481
degenerate_toroidal_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1481
dependent_instantiable_approval_status	
AIM EXPRESS long listing rule .....	1741
dependent_instantiable_approval_status	
AIM EXPRESS short listing rules.....	1338
dependent_instantiable_date	
AIM EXPRESS long listing rule .....	1741
dependent_instantiable_date.....	1339
dependent_instantiable_named_unit	
AIM EXPRESS long listing rule .....	1741
AIM EXPRESS short listing rules.....	1339
dependent_instantiable_precision_qualifier	
AIM EXPRESS long listing rule .....	1741
AIM EXPRESS short listing rules.....	1340
dependent_instantiable_security_classification_level	
AIM EXPRESS long listing rule .....	1742
AIM EXPRESS short listing rules.....	1341



dependent_instantiable_shape_representation	
AIM EXPRESS long listing rule .....	1742
AIM EXPRESS short listing rules .....	1342
dependent_instantiable_type_qualifier	
AIM EXPRESS long listing rule .....	1742
AIM EXPRESS short listing rules .....	1340
dependent_instantiable_uncertainty_qualifier	
AIM EXPRESS long listing rule .....	1742
AIM EXPRESS short listing rules .....	1341
derive_dimensional_exponents	
AIM EXPRESS long listing FUNCTION .....	1756
derived_property_select	
AIM diagram .....	1937
AIM EXPRESS long listing type .....	1401
derived_shape_aspect	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1481
derived_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1482
derived_unit_element	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1482
description_attribute	
AIM diagram .....	1943
AIM EXPRESS long listing entity .....	1482
description_attribute_select	
AIM diagram .....	1943
AIM EXPRESS long listing type .....	1402
Descriptive_parameter	
application assertion .....	300, 301, 305, 307, 317, 320, 321, 326, 336, 337, 339, 351
application object .....	84
ARM diagram .....	1858
mapping specification .....	733
descriptive_representation_item	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1482
mapping specification .....	733, 751, 771
design_phase	
other definitions .....	7
Design_exception	
units of functionality .....	14
Design_exception_notice	
application assertion .....	310
application object .....	85
ARM diagram .....	1849
mapping specification .....	606
design_form	
© ISO 2008 — All rights reserved	2265

ARM diagram .....	1891
design_part	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1482
AIM EXPRESS short listing entity .....	1083
mapping specification .....	849
Design_part	
application assertion .....	310, 334, 339, 342, 370, 377, 378
application object .....	86
ARM diagram .....	1850
mapping specification .....	849
design_reference	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1483
AIM EXPRESS short listing entity .....	1084
mapping specification .....	754
Design_reference	
application assertion .....	296, 310, 311, 343, 379
application object .....	86
ARM diagram .....	1859
mapping specification .....	754
Diameter_dimension_tolerance	
application object .....	88
ARM diagram .....	1877
mapping specification .....	782
die casting	
other definitions .....	7
die_cast_master	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1483
AIM EXPRESS short listing entity .....	1085
mapping specification .....	615
Die_cast_master	
application assertion .....	311
application object .....	88
ARM diagram .....	1871
mapping specification .....	615
die_cast_tooling	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1484
AIM EXPRESS short listing entity .....	1087
mapping specification .....	619
Die_cast_tooling	
application assertion .....	311
application object .....	89
ARM diagram .....	1871
mapping specification .....	619
die_cast_type	

ARM diagram .....	1871
die_clamping	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1485
AIM EXPRESS short listing entity .....	1088
mapping specification .....	622
Die_clamping	
application assertion .....	312, 313, 330, 354
application object .....	90
ARM diagram .....	1872
mapping specification .....	622
die_core	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1485
AIM EXPRESS short listing entity .....	1089
mapping specification .....	623
Die_core	
application assertion .....	312, 355, 356
application object .....	90
ARM diagram .....	1876
mapping specification .....	623
die_core_type	
ARM diagram .....	1876
die_design_feature	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1486
AIM EXPRESS short listing entity .....	1091
mapping specification .....	614
Die_design_feature	
application assertion .....	312
application object .....	92
ARM diagram .....	1863
mapping specification .....	614
die_master	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1487
AIM EXPRESS short listing entity .....	1092
mapping specification .....	628
Die_master	
application assertion .....	312
application object .....	92
ARM diagram .....	1872
mapping specification .....	628
Die_mould_feature	
units of functionality .....	14
Die_mould_requisition	
application object .....	93
ARM diagram .....	1853
© ISO 2008 — All rights reserved	2267

mapping specification.....	954
die_mould_tooling	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1488
AIM EXPRESS short listing entity .....	1094
mapping specification.....	632
Die_mould_tooling	
application assertion .....	311, 313
application object.....	93
ARM diagram .....	1872
mapping specification.....	632
die_mould_vent	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1488
AIM EXPRESS short listing entity .....	1095
mapping specification.....	635
Die_mould_vent	
application assertion .....	346
application object.....	94
ARM diagram .....	1885
mapping specification.....	635
die_select	
ARM diagram.....	1866
Digital_file	
application object.....	94
ARM diagram .....	1900
mapping specification.....	756
Digital_technical_data_package_work_order	
application assertion .....	359
application object.....	95
ARM diagram .....	1853
mapping specification.....	830
dimension_count	
AIM diagram .....	1915
AIM EXPRESS long listing type .....	1402
dimension_of	
AIM EXPRESS long listing FUNCTION .....	1757
dimension_related_tolerance_zone_element	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1489
dimension_type	
ARM diagram.....	1877
dimensional_characteristic	
AIM diagram .....	1926
AIM EXPRESS long listing type .....	1402
dimensional_characteristic_representation	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1489

mapping specification.....	811
dimensional_exponents	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1489
dimensional_location	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1489
mapping specification.....	795
dimensional_location_with_path	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1489
mapping specification.....	786
dimensional_measurement_inspection	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1489
AIM EXPRESS short listing entity .....	1096
mapping specification.....	879
Dimensional_measurement_inspection	
application assertion .....	313
Dimensional_measurement_inspection	
application object.....	95
Dimensional_measurement_inspection	
mapping specification.....	879
Dimensional_measurement_inspection	
ARM diagram .....	1889
dimensional_size	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1490
mapping specification.....	780, 782, 787, 801
dimensional_size_with_path	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1490
mapping specification.....	787, 791, 792, 807, 814
Dimensional_tolerance	
application assertion .....	313
application object.....	95
ARM diagram .....	1877
mapping specification.....	782
dimensional_tolerances_per	
ARM diagram .....	1896
dimensions_for_si_unit	
AIM EXPRESS long listing FUNCTION .....	1758
directed_action	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1490
AIM EXPRESS short listing imported entity modifications .....	1327
mapping specification.....	823, 830, 832, 841
directed_dimensional_location	
© ISO 2008 — All rights reserved	2269

AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1490
AIM EXPRESS short listing entity .....	1097
mapping specification .....	777, 778, 794
Directed_dimensional_location	
mapping specification .....	795
direction	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1490
Direction_element	
application assertion .....	323, 348
application object .....	97
ARM diagram .....	1860
mapping specification .....	961
direction_shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1491
mapping specification .....	961
Distance_along_curve_tolerance	
application assertion .....	314
application object .....	97
ARM diagram .....	1877
mapping specification .....	786
document	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1491
mapping specification .....	608, 753, 765, 770, 771, 772, 775, 954, 955
Document_assignment	
application assertion .....	304, 314, 319, 368, 374
application object .....	99
ARM diagram .....	1899
mapping specification .....	756
document_file	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1491
AIM EXPRESS short listing entity .....	1098
mapping specification .....	756, 757, 768
Document_file	
application assertion .....	310, 314
application object .....	99
ARM diagram .....	1900
mapping specification .....	757
document_file_properties	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1492
AIM EXPRESS short listing entity .....	1099
Document_file_properties	
application assertion .....	314

application object.....	101
ARM diagram.....	1900
mapping specification.....	759
Document_file_relationship	
application assertion.....	314
application object.....	107
ARM diagram.....	1900
mapping specification.....	763
document_file_rules	
AIM EXPRESS long listing rule.....	1742
AIM EXPRESS short listing rules.....	1343
document_product_association	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1493
document_reference	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1493
document_reference_item	
AIM diagram.....	1935
AIM EXPRESS long listing type.....	1402
AIM EXPRESS short listing type.....	1007
document_relationship	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1493
mapping specification.....	763
document_representation_type	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1493
document_type	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1494
ARM diagram.....	1900
document_usage_constraint	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1494
mapping specification.....	774
document_usage_constraint_assignment	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1494
document_usage_role	
AIM diagram.....	1934
AIM EXPRESS long listing entity.....	1494
dot_product	
AIM EXPRESS long listing FUNCTION.....	1758
Draft_angle_surface	
application assertion.....	314, 315
Draft_surface	
application assertion.....	352
© ISO 2008 — All rights reserved	2271

drafted_surface	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1494
AIM EXPRESS short listing entity .....	1101
mapping specification .....	961
Drafted_surface	
application assertion .....	307, 312, 356, 358
application object .....	107
ARM diagram .....	1860
mapping specification .....	961
dryer	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1495
AIM EXPRESS short listing entity .....	1102
mapping specification .....	716
Dryer	
application object .....	108
ARM diagram .....	1895
mapping specification .....	716
dummy_gri	
AIM EXPRESS long listing constant .....	1397
dummy_tri	
AIM EXPRESS long listing constant .....	1397
edge	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1495
edge_curve	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1495
edge_loop	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1495
edge_reversed	
AIM EXPRESS long listing FUNCTION .....	1759
edge_round	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1495
AIM EXPRESS short listing imported entity modifications .....	1327
edge_round_requires_faces	
AIM EXPRESS long listing rule .....	1744
AIM EXPRESS short listing rules .....	1345
Edge_shape_element	
application object .....	108
ARM diagram .....	1860
mapping specification .....	962
edge_shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1497



AIM EXPRESS short listing entity .....	1103
mapping specification .....	962
effectivity	
AIM diagram .....	1906
AIM EXPRESS long listing entity .....	1497
effectivity_assignment	
AIM diagram .....	1906
AIM EXPRESS long listing entity .....	1497
effectivity_item	
AIM diagram .....	1906
AIM EXPRESS long listing type .....	1402
AIM EXPRESS short listing type .....	1008
ejector_box	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1498
AIM EXPRESS short listing entity .....	1103
Ejector_box	
application assertion .....	315
application object .....	108
ARM diagram .....	1866
mapping specification .....	403
ejector_design_feature	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1499
AIM EXPRESS short listing entity .....	1105
Ejector_design_feature	
application assertion .....	316
ARM diagram .....	1863
Ejector_designj_feature	
application object .....	109
ejector_equipment	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1499
AIM EXPRESS short listing entity .....	1106
mapping specification .....	717
Ejector_equipment	
application assertion .....	306, 316
application object .....	109
ARM diagram .....	1895
mapping specification .....	717
ejector_pin	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1499
AIM EXPRESS short listing entity .....	1106
Ejector_pins	
application assertion .....	313, 316
application object .....	110
ARM diagram .....	1866
© ISO 2008 — All rights reserved	2273

mapping specification.....	410
ejector_plate	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1500
AIM EXPRESS short listing entity .....	1108
Ejector_plate	
application assertion .....	315, 316, 317
application object.....	110
ARM diagram .....	1866
mapping specification.....	413
ejector_system	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1501
AIM EXPRESS short listing entity .....	1110
Ejector_system	
application assertion .....	316, 317, 347, 356
application object.....	112
ARM diagram .....	1866
mapping specification.....	418
ejector_types	
ARM diagram .....	1866
electric_current_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1403
elementary_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1502
elevated_temperature_service	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1502
AIM EXPRESS short listing entity .....	1111
elevated_Temperature_Service	
mapping specification.....	569
Elevated_temperature_service	
application assertion .....	301, 317
application object.....	112
ARM diagram .....	1898
Elevated_Temperature_Service	
mapping specification.....	569
ellipse	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1503
Engineering_change_order	
application assertion .....	317
application object.....	113
ARM diagram .....	1849
mapping specification.....	607
Engineering_change_proposal	

application assertion .....	310, 317
application object.....	114
ARM diagram .....	1849
mapping specification.....	608
Equipment	
application assertion .....	302, 317, 318, 319, 366
application object.....	114
ARM diagram .....	1894
mapping specification.....	718
Equipment_process	
application assertion .....	318
application object.....	115
ARM diagram .....	1888
mapping specification.....	880
equipment_setting_record	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1503
AIM EXPRESS short listing entity .....	1113
mapping specification.....	935
Equipment_setting_record	
application object.....	116
ARM diagram .....	1883
mapping specification.....	935
equipment_setup	
mapping specification.....	906
Equipment_setup	
application assertion .....	299, 318
application object.....	116
ARM diagram .....	1894
mapping specification.....	881
Equipment_size	
application assertion .....	306, 318, 332
application object.....	117
ARM diagram .....	1894
mapping specification.....	720
equipment_type	
ARM diagram .....	1894
equipment_usage	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1504
AIM EXPRESS short listing entity .....	1113
mapping specification.....	936
Equipment_usage	
application assertion .....	318, 319
application object.....	118
ARM diagram .....	1894
mapping specification.....	936
Executable	
© ISO 2008 — All rights reserved	2275

application object.....	118
ARM diagram.....	1884
mapping specification.....	763
executed_action	
AIM diagram.....	1927
AIM EXPRESS long listing entity.....	1505
extension	
AIM diagram.....	1925
AIM EXPRESS long listing entity.....	1505
External_file_identification	
application assertion.....	314
application object.....	119
ARM diagram.....	1900
mapping specification.....	764
external_identification_assignment	
AIM diagram.....	1936
AIM EXPRESS long listing entity.....	1505
external_identification_item	
AIM diagram.....	1936
AIM EXPRESS long listing type.....	1403
AIM EXPRESS short listing type.....	1008
External_schema_definition	
application assertion.....	319
External_schema_definition	
application assertion.....	298, 313
application object.....	119
External_schema_definition	
application assertion.....	332
External_schema_definition	
application assertion.....	334
External_schema_definition	
application assertion.....	343
External_schema_definition	
mapping specification.....	764
External_schema_definition	
ARM diagram.....	1884
external_source	
AIM diagram.....	1936
AIM EXPRESS long listing entity.....	1505
externally_defined_class	
AIM diagram.....	1935
AIM EXPRESS long listing entity.....	1505
AIM EXPRESS short listing entity.....	1114
mapping specification.....	698, 699
externally_defined_class_with_known_source_requirement	
AIM EXPRESS long listing rule.....	1745
AIM EXPRESS short listing rules.....	1347
externally_defined_dimension_definition	

AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1506
AIM EXPRESS short listing entity .....	1115
mapping specification .....	787
externally_defined_feature_definition	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1506
externally_defined_feature_definiton	
AIM EXPRESS short listing imported entity modifications .....	1327
externally_defined_general_property	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1513
AIM EXPRESS short listing entity .....	1116
mapping specification .....	703
externally_defined_item	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1513
externally_defined_item_relationship	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1513
externally_defined_representation	
mapping specification .....	700
Externally_defined_representation	
application assertion .....	319, 364
application object .....	120
ARM diagram .....	1901
mapping specification .....	700
externally_defined_representation_with_parameter	
AIM EXPRESS short listing entity .....	1117
externally_defined_representation_with_parameters	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1514
externally_defined_schema	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1514
AIM EXPRESS short listing entity .....	1118
mapping specification .....	763, 764
Externally_defined_size_dimension	
application assertion .....	319
application object .....	121
ARM diagram .....	1877
mapping specification .....	787
fabrication	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1514
AIM EXPRESS short listing entity .....	1119
mapping specification .....	881
Fabrication	
© ISO 2008 — All rights reserved	2277

application object.....	121
ARM diagram.....	1886
mapping specification.....	881
face	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1514
face_bound	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1515
face_bound_reversed	
AIM EXPRESS long listing FUNCTION.....	1759
face_outer_bound	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1515
face_reversed	
AIM EXPRESS long listing FUNCTION.....	1760
Face_shame_element	
application assertion.....	323
Face_shape_element	
application assertion.....	314, 319, 344, 349
application object.....	121
ARM diagram.....	1860
mapping specification.....	963
Face_shape_element_relationship	
application assertion.....	319
application object.....	122
ARM diagram.....	1860
mapping specification.....	963
face_shape_representation	
AIM diagram.....	1913
AIM EXPRESS long listing entity.....	1515
mapping specification.....	963
face_shape_representation_relationship	
AIM diagram.....	1912
AIM EXPRESS long listing entity.....	1515
AIM EXPRESS short listing entity.....	1119
face_shape_representation_representation	
mapping specification.....	963
face_surface	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1515
faceted_brep	
AIM diagram.....	1923
AIM EXPRESS long listing entity.....	1516
Feature	
application assertion.....	362
application object.....	122
ARM diagram.....	1861

mapping specification.....	819
feature_component_definition	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1516
feature_component_relationship	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1516
feature_definition	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1516
mapping specification.....	820
mapping table .....	622, 623
Feature_definition	
mapping specification.....	815, 816, 817, 818
feature_pattern	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1517
feeder	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1518
AIM EXPRESS short listing entity .....	1120
Feeder	
application assertion .....	365
application object.....	122
ARM diagram.....	1864
mapping specification.....	420
Feeders	
application assertion .....	320
fillet	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1519
filter	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1520
AIM EXPRESS short listing entity .....	1122
Filter	
application assertion .....	320, 365, 374
application object.....	123
ARM diagram.....	1864
mapping specification.....	423
Finishing	
application object.....	124
ARM diagram.....	1894
mapping specification.....	722
finishing_and_machining_operation	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1521
AIM EXPRESS short listing entity .....	1123
© ISO 2008 — All rights reserved	2279

finishing_equipment	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1522
AIM EXPRESS short listing entity .....	1124
mapping specification .....	722
finishing_or_machining_operations	
mapping specification .....	571
Finishing_or_machining_operations	
application assertion .....	301, 320, 339
application object .....	124
ARM diagram .....	1898
mapping specification .....	571
first_article	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1522
AIM EXPRESS short listing entity .....	1124
mapping specification .....	936
First_article	
application assertion .....	308, 320, 321
application object .....	125
ARM diagram .....	1889
mapping specification .....	936
first_proj_axis	
AIM EXPRESS long listing FUNCTION .....	1760
fixture_tab	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1522
AIM EXPRESS short listing entity .....	1126
Fixturing_tab	
ARM diagram .....	1862
Fixturing_tab	
mapping specification .....	425
flask	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1523
AIM EXPRESS short listing entity .....	1126
Flask	
application assertion .....	321, 331, 339, 345
application object .....	126
ARM diagram .....	1869
mapping specification .....	426
Flask application assertion .....	345
flask_select	
ARM diagram .....	1885
flaskless	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1524
AIM EXPRESS short listing entity .....	1129



Flaskless	
application assertion .....	321, 340, 345
application object.....	128
ARM diagram .....	1885
mapping specification .....	432
flat_face	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1525
flatness_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1526
mapping specification .....	788
flow_select	
ARM diagram .....	1885
founded_item	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1527
founded_item_select	
AIM diagram .....	1918
AIM EXPRESS long listing type .....	1403
foundry representative	
other definitions .....	7
foundry_to_specify_for	
ARM diagram .....	1896
functionally_defined_transformation	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1527
gating_component_select	
ARM diagram .....	1864
gating_design_feature	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1527
AIM EXPRESS short listing entity .....	1130
Gating_design_feature	
application assertion .....	321, 331
application object.....	129
ARM diagram .....	1863
mapping specification .....	409, 434, 541
gating_system	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1527
AIM EXPRESS short listing entity .....	1131
Gating_system	
application assertion .....	298, 322, 338, 346, 355, 378
application object.....	129
ARM diagram .....	1864
mapping specification.....	436
gear	

AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1528
General_foundry_process	
application assertion .....	299, 300
application object.....	45
ARM diagram .....	1884
mapping specification.....	874
general_property	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1532
general_property_association	
AIM diagram .....	1937
AIM EXPRESS long listing entity .....	1532
General_rib	
application assertion .....	322, 323
application object.....	129
ARM diagram .....	1862
mapping specification.....	472
generic_casting_resource	
mapping specification.....	722
Generic_casting_resource	
application assertion .....	296
ARM diagram .....	1882
mapping specification.....	722
generic_manufacturing_resource	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1532
AIM EXPRESS short listing entity .....	1132
Generic_manufacturing_resource	
application object.....	130
geometric_alignment	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1533
Geometric_characteristics	
mapping specification.....	733
geometric_intersection	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1533
Geometric_model	
application assertion .....	338, 341, 367
application object.....	131
ARM diagram .....	1859
mapping specification.....	964
geometric_representation_context	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1533
geometric_representation_item	
AIM diagram .....	1915

AIM EXPRESS long listing entity .....	1533
mapping specification .....	964
geometric_set_select	
AIM diagram .....	1915
AIM EXPRESS long listing type .....	1403
geometric_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1534
mapping specification .....	788
Geometric_tolerance	
application assertion .....	323
application object .....	131
ARM diagram .....	1879
mapping specification .....	788
geometric_tolerance_precedence_relationship	
mapping specification .....	790
Geometric_tolerance_precedence_relationship	
application assertion .....	323
application object .....	133
ARM diagram .....	1879
mapping specification .....	790
geometric_tolerance_relationship	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1534
geometric_tolerance_with_datum_reference	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1534
geometric_tolerance_with_defined_unit	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1534
Geometry_characteristic	
application assertion .....	323, 324, 335
application object .....	133
get_action_property_representation	
AIM EXPRESS long listing FUNCTION .....	1778
AIM EXPRESS short listing function .....	1360
get_basis_surface	
AIM EXPRESS long listing FUNCTION .....	1761
get_description_value	
AIM EXPRESS long listing FUNCTION .....	1762
get_id_value	
AIM EXPRESS long listing FUNCTION .....	1762
get_name_value	
AIM EXPRESS long listing FUNCTION .....	1762
get_property_definition_representations	
AIM EXPRESS long listing FUNCTION .....	1763
get_property_definition_shape_aspect	
AIM EXPRESS long listing FUNCTION .....	1778
© ISO 2008 — All rights reserved	2283

AIM EXPRESS short listing function .....	1359
get_resource_property_representation	
AIM EXPRESS long listing FUNCTION .....	1779
AIM EXPRESS short listing function .....	1360
get_role	
AIM EXPRESS long listing FUNCTION .....	1763
global_uncertainty_assigned_context	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1534
global_unit_assigned_context	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1534
graphics_type_select	
ARM diagram .....	1881
green sand	
other definitions .....	7
group	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1535
mapping specification .....	821
group_assignment	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1535
group_item	
AIM diagram .....	1935
AIM EXPRESS long listing type .....	1403
AIM EXPRESS short listing type .....	1008
group_relationship	
AIM diagram .....	1935
AIM EXPRESS long listing entity .....	1535
Hardcopy	
application object .....	134
ARM diagram .....	1900
mapping specification .....	768
Hardness	
application assertion .....	335, 336
application object .....	134
ARM diagram .....	1856
mapping specification .....	734
Heat	
application assertion .....	324, 333, 366
application object .....	135
ARM diagram .....	1851
mapping specification .....	939
heat_treat	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1535
AIM EXPRESS short listing entity .....	1133

mapping specification.....	882
Heat_treat	
application assertion .....	324
application object.....	135
ARM diagram.....	1886
mapping specification.....	882
heat_treat_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1535
AIM EXPRESS short listing entity .....	1133
mapping specification.....	572
Heat_treat_requirement	
application assertion .....	324
application object.....	136
ARM diagram.....	1888
mapping specification.....	572
Height_dimension	
application assertion .....	324, 791
application object.....	137
ARM diagram.....	1877
hole_bottom	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1537
hour_in_day	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1403
hyperbola	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1540
id_attribute	
AIM diagram .....	1943
AIM EXPRESS long listing entity .....	1540
id_attribute_select	
AIM diagram .....	1943
AIM EXPRESS long listing type .....	1404
identification_assignment	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1540
identification_assignment_item	
AIM diagram .....	1936
AIM EXPRESS long listing type .....	1404
identification_item	
AIM EXPRESS short listing type .....	1008
identification_role	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1540
identifier	
AIM diagram .....	1945
© ISO 2008 — All rights reserved	2285

AIM EXPRESS long listing type .....	1404
illustration	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1541
AIM EXPRESS short listing entity .....	1136
mapping specification .....	768
Illustration	
application assertion .....	296, 307, 324, 335, 371
application object .....	138
ARM diagram .....	1859
mapping specification .....	768
implicit_or_explicit_shape	
ARM diagram .....	1865
Implementation method .....	5
impregnation_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1541
AIM EXPRESS short listing entity .....	1136
mapping specification .....	882
Impregnation_activity	
application object .....	138
ARM diagram .....	1886
mapping specification .....	882
in_facility_location	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1541
AIM EXPRESS short listing entity .....	1137
mapping specification .....	724
In_facility_location	
application assertion .....	311, 313, 318, 331, 345, 352, 355, 358
application object .....	138
ARM diagram .....	1894
mapping specification .....	724
Indirect_stock_requisition	
application object .....	139
ARM diagram .....	1853
mapping specification .....	955
ingate	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1541
AIM EXPRESS short listing entity .....	1137
Ingate	
application assertion .....	322, 325, 347, 374
application object .....	141
ARM diagram .....	1864
ingate_contact	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1542

AIM EXPRESS short listing entity .....	1139
Ingate_contacts	
application assertion .....	325
application object.....	139
ARM diagram.....	1864
mapping specification.....	443
Ingates	
mapping specification.....	440
insert_type	
ARM diagram.....	1871
inspection_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1544
AIM EXPRESS short listing entity .....	1142
mapping specification.....	883
Inspection_activity	
application assertion .....	325, 326, 366
application object.....	142
ARM diagram.....	1889
mapping specification.....	883
inspection_or_test_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1546
AIM EXPRESS short listing entity .....	1145
mapping specification.....	575, 579
Inspection_or_test_requirement	
application assertion .....	326, 341
application object.....	144
ARM diagram.....	1890
mapping specification.....	575
inspection_or_test_result	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1547
AIM EXPRESS short listing entity .....	1147
mapping specification.....	939
Inspection_or_test_result	
application assertion .....	327, 351
application object.....	146
ARM diagram.....	1852
mapping specification.....	939
inspection_plan	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1548
AIM EXPRESS short listing entity .....	1148
Inspection_plan	
application assertion .....	308, 327
application object.....	147
ARM diagram.....	1890
© ISO 2008 — All rights reserved	2287

ISO 10303-223:2008 (E)

mapping specification.....	579
inspection_procedures_types	
ARM diagram.....	1890
Inspection_reason	
ARM diagram.....	1889
inspection_requirement_type	
ARM diagram.....	1890
Inspection_type	
ARM diagram.....	1889
instanced_feature	
AIM diagram.....	1903, 1909
AIM EXPRESS long listing entity.....	1549
AIM EXPRESS short listing imported entity modifications.....	1327
mapping specification.....	818, 819, 820
Instanced_feature	
mapping specification.....	816, 818
mapping table.....	821
Integrated_resource.....	5
integration_interval	
AIM diagram.....	1928
AIM EXPRESS long listing entity.....	1549
AIM EXPRESS short listing entity.....	1150
mapping specification.....	977
Integration_interval	
application assertion.....	327, 368
application object.....	148
ARM diagram.....	1892
mapping specification.....	977
investment_casting	
other definitions.....	7
Investment_casting_features	
units of functionality.....	15
investment_casting_master	
AIM diagram.....	1905
AIM EXPRESS long listing entity.....	1549
AIM EXPRESS short listing entity.....	1151
mapping specification.....	678
Investment_casting_master	
application object.....	149
ARM diagram.....	1873
mapping specification.....	678
investment_casting_tooling	
AIM diagram.....	1905
AIM EXPRESS long listing entity.....	1550
AIM EXPRESS short listing entity.....	1152
mapping specification.....	679, 683
Investment_casting_tooling	
application assertion.....	328



Investment_casting_tooling	
application object.....	149
Investment_casting_tooling	
mapping specification.....	679
Investment_casting_tooling	
mapping specification.....	683
Investment_casting_tooling	
ARM diagram.....	1873
investment_casting_tooling_definition	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1551
AIM EXPRESS short listing entity.....	1154
Investment_casting_tooling_definition	
application assertion.....	328, 329
application object.....	150
ARM diagram.....	1873
investment_design_feature	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1553
AIM EXPRESS short listing entity.....	1157
Investment_design_feature	
application assertion.....	329
application object.....	151
ARM diagram.....	1863
mapping specification.....	689
investment_mould	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1553
AIM EXPRESS short listing entity.....	1158
mapping specification.....	690
Investment_mould	
application assertion.....	329, 340, 374
application object.....	151
ARM diagram.....	1873
mapping specification.....	690
item_in_context	
AIM EXPRESS long listing FUNCTION.....	1763
Item_size	
application assertion.....	302, 305, 307, 312, 315, 316, 320, 325, 329, 331, 338, 339, 346, 352, 353, 355, 356, 364, 365, 368, 374, 375, 378, 379
application object.....	152
ARM diagram.....	1865
mapping specification.....	449
knot_type	
AIM diagram.....	1919
AIM EXPRESS long listing type.....	1404
known_source	
AIM diagram.....	1936
© ISO 2008 — All rights reserved	2289

ISO 10303-223:2008 (E)

AIM EXPRESS long listing entity .....	1553
AIM EXPRESS short listing entity .....	1159
label	
AIM diagram .....	1945
AIM EXPRESS long listing type .....	1404
ARM diagram .....	1858
leap_year	
AIM EXPRESS long listing FUNCTION .....	1764
Length_dimension	
application assertion .....	329, 792
application object .....	153
ARM diagram .....	1877
length_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1404
length_measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1554
length_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1554
library_class_version_assignment	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1554
AIM EXPRESS short listing entity .....	1160
Library_part_assignment	
application assertion .....	319, 330
application object .....	154
ARM diagram .....	1901
mapping specification .....	701
library_property_version_assignment	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1554
AIM EXPRESS short listing entity .....	1160
Library_reference	
units of functionality .....	16
limit_condition	
AIM diagram .....	1924
AIM EXPRESS long listing type .....	1404
mapping specification .....	796
limit_type	
ARM diagram .....	1877
limits_and_fits	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1554
mapping specification .....	792
Limits_and_fits	
application assertion .....	341

Limits_and_fits	
application object.....	154
Limits_and_fits	
application assertion .....	376
Limits_and_fits	
mapping specification.....	792
Limits_and_fits	
ARM diagram.....	1877
line	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1554
line_profile_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1555
mapping specification.....	788
linear_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1555
list_face_loops	
AIM EXPRESS long listing FUNCTION .....	1764
list_of_reversible_topology_item	
AIM diagram .....	1922
AIM EXPRESS long listing type .....	1405
list_of_topology_reversed	
AIM EXPRESS long listing FUNCTION .....	1765
list_to_array	
AIM EXPRESS long listing FUNCTION .....	1765
list_to_set	
AIM EXPRESS long listing FUNCTION .....	1765
local_time	
AIM diagram .....	1938
AIM EXPRESS long listing entity .....	1556
Location_dimension_tolerance	
application assertion .....	297, 330
application object.....	155
ARM diagram.....	1877
mapping specification.....	794
Location_element	
application assertion .....	302, 310, 332, 346, 348, 372, 378
application object.....	156
ARM diagram.....	1860
mapping specification.....	964
location_shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1556
mapping specification.....	964
Location_tolerance	
application assertion .....	330
© ISO 2008 — All rights reserved	2291

ISO 10303-223:2008 (E)

application object.....	156
ARM diagram.....	1877
mapping specification.....	795
loop	
AIM diagram.....	1922
AIM EXPRESS long listing entity.....	1556
loose_piece	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1556
AIM EXPRESS short listing entity.....	1160
Loose_piece	
application assertion.....	330, 352, 358
application object.....	159
ARM diagram.....	1867
mapping specification.....	454
lost_foam_casting_die	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1557
AIM EXPRESS short listing entity.....	1161
Lost_foam_casting_die	
application assertion.....	328, 330, 331, 332, 340
application object.....	157
ARM diagram.....	1874
Lot	
application assertion.....	332, 333, 366
application object.....	160
ARM diagram.....	1851
mapping specification.....	941
lot_effectivity	
AIM diagram.....	1906
AIM EXPRESS long listing entity.....	1560
luminous_intensity_measure	
AIM diagram.....	1942
AIM EXPRESS long listing type.....	1405
machine	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1561
AIM EXPRESS short listing entity.....	1167
machine_allowance	
AIM diagram.....	1907
AIM EXPRESS long listing entity.....	1561
AIM EXPRESS short listing entity.....	1168
Machine_design_feature	
application assertion.....	310
machine_element_relationship	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1561
AIM EXPRESS short listing entity.....	1169

machine_mounting	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1562
AIM EXPRESS short listing entity .....	1169
mapping specification .....	725
Machine_mounting	
application assertion .....	306, 331, 332
application object .....	160
ARM diagram .....	1895
mapping specification .....	725
machine_process	
AIM EXPRESS short listing entity .....	1167
machine_setup	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1561
AIM EXPRESS short listing entity .....	1168
machining_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1563
AIM EXPRESS short listing entity .....	1172
mapping specification .....	888
Machining_activity	
application assertion .....	320, 332
application object .....	162
ARM diagram .....	1886
mapping specification .....	888
machining_allowance	
mapping specification .....	581
Machining_allowance	
application assertion .....	298, 311, 312, 332, 333, 352, 356
application object .....	162
ARM diagram .....	1862
mapping specification .....	581
Machining_feature	
application assertion .....	303, 316, 332
application object .....	163
ARM diagram .....	1861
mapping specification .....	815, 820
Machining_feature_group	
mapping specification .....	821
machining_feature_life_cycle	
AIM EXPRESS long listing rule .....	1745
Machining_feature_life_cycle	
AIM EXPRESS short listing rules .....	1347
machining_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1561
make_array_of_array	
© ISO 2008 — All rights reserved	2293

AIM EXPRESS long listing FUNCTION .....	1766
make_from_usage_option	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1563
manifold_solid_brep	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1564
manufacturing_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1564
AIM EXPRESS short listing entity .....	1172
manufacturing_activity_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1564
AIM EXPRESS short listing entity .....	1173
Manufacturing_casting_resources	
units of functionality .....	16
Manufacturing_design_feature	
application object .....	163
ARM diagram .....	1861
mapping specification .....	820
Manufacturing_feature	
application assertion .....	332, 333
application object .....	164
ARM diagram .....	1861
mapping specification .....	819
units of functionality .....	18
Manufacturing_feature_group	
application assertion .....	333
application object .....	164
ARM diagram .....	1861
manufacturing_group_select	
ARM diagram .....	1861
Manufacturing_part_properties	
units of functionality .....	18
manufacturing_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1564
AIM EXPRESS short listing entity .....	1173
mapping specification .....	889
Manufacturing_process	
application assertion .....	305, 333, 350
application object .....	165
ARM diagram .....	1888
mapping specification .....	889
Manufacturing_process_control_documentation	
units of functionality .....	21
manufacturing_process_relationship	

AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1565
AIM EXPRESS short listing entity .....	1174
Manufacturing_process_requirement_documents	
units of functionality .....	19
mapped_item	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1565
marking	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1565
mass_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1405
mass_measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1567
mass_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1567
master_pattern	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1567
AIM EXPRESS short listing entity .....	1175
Master_pattern	
application assertion .....	333
application object .....	166
ARM diagram .....	1867
mapping specification .....	456
master_sample	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1568
AIM EXPRESS short listing entity .....	1176
mapping specification .....	850
Master_sample	
application assertion .....	321, 333, 334, 366
application object .....	166
ARM diagram .....	1851
mapping specification .....	850
Material	
application assertion .....	303, 311, 313, 334, 336, 343, 353, 362, 364
application object .....	167
ARM diagram .....	1857
mapping specification .....	737
Material_composition	
application assertion .....	334, 335
ARM diagram .....	1857
mapping specification .....	738
© ISO 2008 — All rights reserved	2295

Material_composition_modifier	
application object.....	169
Material_condition_modifier	
application assertion .....	309, 323
application object.....	171
ARM diagram .....	1879
mapping specification.....	796
material_designation	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1569
material_is_specified_for_part	
AIM EXPRESS long listing rule .....	1745
AIM EXPRESS short listing rules.....	1348
material_property	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1569
mapping specification.....	740
Material_property	
application assertion .....	303, 334, 335, 336, 360
application object.....	171
ARM diagram.....	1856
mapping specification.....	740
material_property_representation	
AIM diagram .....	1908
AIM EXPRESS long listing entity .....	1569
mapping specification.....	734
Material_requisition	
application object.....	172
ARM diagram .....	1853
mapping specification.....	955
material_structure	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1569
AIM EXPRESS short listing entity .....	1178
mapping specification.....	741
Material_structure	
application assertion .....	334, 335
application object.....	172
ARM diagram.....	1857
mapping specification.....	741
Material_structure_element	
application assertion .....	335, 336
application object.....	173
ARM diagram .....	1857
mapping specification.....	743
material_usage	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1570



AIM EXPRESS short listing entity .....	1179
mapping specification .....	942
Material_usage	
application assertion .....	336, 350
application object .....	173
ARM diagram .....	1857
mapping specification .....	942
material_usage_record	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1570
AIM EXPRESS short listing entity .....	1180
mapping specification .....	942
Material_usage_record	
application assertion .....	336
application object .....	174
ARM diagram .....	1883
mapping specification .....	942
measure_qualification	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1571
measure_representation_item	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1571
mapping specification .....	744, 746, 747, 751
measure_value	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1405
measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1571
Measurement_limitations	
units of functionality .....	21
mechanical part	
other definitions .....	7
Mechanical_property_requirement	
application assertion .....	324
mechanical_property_requirements	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1571
AIM EXPRESS short listing entity .....	1181
mapping specification .....	582
Mechanical_property_requirements	
application assertion .....	336, 337
application object .....	174
ARM diagram .....	1888
mapping specification .....	582
mechanical_stress	
AIM diagram .....	1907
© ISO 2008 — All rights reserved	2297

AIM EXPRESS long listing entity .....	1573
AIM EXPRESS short listing entity .....	1183
mapping specification .....	585
Mechanical_stress	
application .....	175
application assertion .....	301, 337
ARM diagram .....	1898
mapping specification .....	585
melting_equipment	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1573
AIM EXPRESS short listing entity .....	1184
mapping specification .....	727
Melting_equipment	
application object .....	176
ARM diagram .....	1894
mapping specification .....	727
meshing_condition	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1573
AIM EXPRESS short listing entity .....	1186
mapping specification .....	979
Meshing_condition	
application assertion .....	337, 369
application object .....	176
ARM diagram .....	1892
mapping specification .....	979
Metal_caster_to_simulation	
application object .....	177
metalcaster_simulation	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1575
AIM EXPRESS short listing entity .....	1184
Metalcaster_simulation	
application assertion .....	337, 338
ARM diagram .....	1891
mapping specification .....	890
minute_in_hour	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1405
mismatch .....	5
mismatch_tolerance	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1576
AIM EXPRESS short listing entity .....	1189
mapping specification .....	587
Mismatch_tolerance	
application assertion .....	299, 338, 375

application object.....	178
ARM diagram.....	1860
mapping specification.....	587
mixed_loop_type_set	
AIM EXPRESS long listing FUNCTION.....	1766
Model.....	5
Model_element	
application assertion.....	338, 367
application object.....	179
ARM diagram.....	1859
mapping specification.....	964
modified_part_design	
other definitions.....	8
modified_geometric_tolerance	
AIM diagram.....	1924
AIM EXPRESS long listing entity.....	1576
modified_pattern	
AIM diagram.....	1911
AIM EXPRESS long listing entity.....	1576
month_in_year_number	
AIM diagram.....	1938
AIM EXPRESS long listing type.....	1405
mould	
other definitions.....	8
mould_box	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1577
AIM EXPRESS short listing entity.....	1189
Mould_box	
application assertion.....	321, 338, 339
ARM diagram.....	1885
mapping specification.....	458
Mould_boxes	
application object.....	179
mould_lock	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1578
AIM EXPRESS short listing entity.....	1190
Mould_locks	
application assertion.....	339, 346
application object.....	180
ARM diagram.....	1868
mapping specification.....	460
moulding_equipment	
AIM diagram.....	1933
AIM EXPRESS long listing entity.....	1578
AIM EXPRESS short listing entity.....	1191
mapping specification.....	728
© ISO 2008 — All rights reserved	2299

Moulding_equipment	
application assertion .....	346
application object.....	180
ARM diagram .....	1894
mapping specification.....	728
mount_type	
ARM diagram .....	1895
movable_slide_select	
ARM diagram .....	1872
msb_shells	
AIM EXPRESS long listing FUNCTION .....	1767
name_attribute	
AIM diagram .....	1944
AIM EXPRESS long listing entity .....	1578
name_attribute_select	
AIM diagram .....	1944
AIM EXPRESS long listing type .....	1406
named_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1578
next_assembly_usage_occurrence	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1579
ngon_closed_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1579
non_casting_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1580
AIM EXPRESS short listing entity .....	1192
non_casting_equipment_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1581
AIM EXPRESS short listing entity .....	1193
mapping specification.....	894
non_casting_operation	
mapping specification.....	892
Non_casting_operation	
mapping specification.....	892
Non_casting_operations	
application assertion .....	339
application object.....	181
ARM diagram .....	1886
Non_equipment_process	
application object.....	182
ARM diagram .....	1888
mapping specification.....	894
non_machining_process	

AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1581
AIM EXPRESS short listing entity .....	1193
non_permanent_moulding_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1581
AIM EXPRESS short listing entity .....	1194
mapping specification .....	895
Non_permanent_moulding_process	
application assertion .....	339, 340
application object .....	182
ARM diagram .....	1885
mapping specification .....	895
Non_permanent_moulding_process_type	
ARM diagram .....	1885
normalise	
AIM EXPRESS long listing FUNCTION .....	1767
Numeric_parameter	
application assertion 298, 299, 300, 303, 304, 307, 314, 316, 317, 318, 321, 322, 326, 327, 329, 332, 335, 336, 337, 338, 344, 348, 369, 375	
application object .....	184
ARM diagram .....	1858
mapping specification .....	744
Numeric_parameter_with_tolerance	
application assertion .....	340, 341, 348
application object .....	185
ARM diagram .....	1858
mapping specification .....	746
Numeric_range	
application assertion .....	323, 336, 337, 341
application object .....	185
ARM diagram .....	1858
mapping specification .....	747
Object_element_shape_representation	
application assertion .....	341, 342
application object .....	186
ARM diagram .....	1859
mapping specification .....	965
object_role	
AIM diagram .....	1944
AIM EXPRESS long listing entity .....	1583
open_path_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1583
open_shell	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1584
open_shell_reversed	
© ISO 2008 — All rights reserved	2301

AIM EXPRESS long listing FUNCTION .....	1768
ordered_item	
AIM diagram .....	1927
AIM EXPRESS long listing type .....	1406
AIM EXPRESS short listing type .....	1009
ordered_part	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1584
AIM EXPRESS short listing entity .....	1196
mapping specification .....	831
Ordered_part	
application assertion .....	308, 343, 350
application object .....	186
ARM diagram .....	1855
mapping specification .....	831
organization	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1585
mapping specification .....	706, 844
Organization	
application assertion .....	296, 341, 343, 347, 373
application object .....	187
ARM diagram .....	1854
mapping specification .....	844
organization_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1585
organization_item	
AIM diagram .....	1939
AIM EXPRESS long listing type .....	1406
AIM EXPRESS short listing type .....	1009
organization_role	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1585
organizational_address	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1585
organizational_project	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1585
Orientation	
application assertion .....	297, 299, 312, 316, 321, 329, 330, 332, 348, 366
application object .....	188
ARM diagram .....	1860
mapping specification .....	965
oriented_closed_shell	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1585

oriented_edge	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1586
oriented_face	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1586
oriented_open_shell	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1586
oriented_path	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1587
oriented_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1587
orthogonal_complement	
AIM EXPRESS long listing FUNCTION .....	1768
other_inspection	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1587
AIM EXPRESS short listing entity .....	1197
mapping specification .....	950
Other_inspection	
application assertion .....	341
application object .....	188
ARM diagram .....	1852
mapping specification .....	950
outer_round	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1588
outside_profile	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1590
parabola	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1594
parallel_offset	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1594
parallelism_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1595
mapping specification .....	788
parameter_value	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1406
parametric_representation_context	
AIM diagram .....	1912
© ISO 2008 — All rights reserved	2303

AIM EXPRESS long listing entity .....	1595
Part administration data	
units of functionality .....	23
Part model	
units of functionality .....	24
Part_dimensioning_standard	
application assertion .....	342
application object .....	189
ARM diagram .....	1899
mapping specification .....	769
Part_property	
application assertion .....	342, 360
application object .....	189
ARM diagram .....	1856
mapping specification .....	747
Part_relationship	
application assertion .....	342
application object .....	189
ARM diagram .....	1850
mapping specification .....	852
part_requires_project_order	
AIM EXPRESS long listing rule .....	1745
AIM EXPRESS short listing rules .....	1349
part_requires_security_classification	
AIM EXPRESS long listing rule .....	1746
AIM EXPRESS short listing rules .....	1350
part_select	
ARM diagram .....	1850
Part_shape	
application assertion .....	342, 343, 344
application object .....	190
ARM diagram .....	1859
mapping specification .....	966
part_to_approval	
AIM EXPRESS long listing rule .....	1746
AIM EXPRESS short listing rules .....	1349
Part_version	
application assertion .....	310, 311, 317, 326, 342, 343, 344, 350, 359, 363, 369
application object .....	191
ARM diagram .....	1849
mapping specification .....	853
partial_circular_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1595
parting_surface	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1596
AIM EXPRESS short listing entity .....	1198



mapping specification.....	968
Parting_surface	
application assertion .....	312, 338, 344, 345, 346, 353, 356, 358, 379
application object.....	194
ARM diagram.....	1860
mapping specification.....	968
path	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1597
Path_element	
application object.....	195
ARM diagram.....	1860
mapping specification.....	970
path_feature_component	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1597
path_head_to_tail	
AIM EXPRESS long listing FUNCTION .....	1769
path_reverse	
AIM EXPRESS long listing FUNCTION .....	1769
path_shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1599
mapping specification.....	970
pattern	
other definitions.....	8
pattern_condition	
ARM diagram.....	1867
pattern_equipment_available	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1600
AIM EXPRESS short listing entity .....	1200
mapping specification.....	588
Pattern_equipment_available	
application assertion .....	338
application object.....	195
ARM diagram.....	1897
mapping specification.....	588
pattern_offset_membership	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1604
pattern_omit_membership	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1607
pattern_plate	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1609
AIM EXPRESS short listing entity .....	1206
© ISO 2008 — All rights reserved	2305

Pattern_plate	
application assertion .....	345, 346, 359
application object.....	199
ARM diagram .....	1868
mapping specification.....	462
pattern_plate_type	
ARM diagram .....	1868
Pattern_tooling_requisition	
application object.....	202
ARM diagram .....	1853
mapping specification.....	955
pattern_tooling_select	
ARM diagram .....	1885
pattern_type	
ARM diagram .....	1897
payment_and_shipping	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1611
AIM EXPRESS short listing entity .....	1210
mapping specification.....	598
Payment_and_shipping	
application object.....	202
ARM diagram .....	1896
mapping specification.....	598
pcurve	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1611
pcurve_or_surface	
AIM diagram .....	1917
AIM EXPRESS long listing type .....	1406
Pedigree_creation_order	
application assertion .....	360
application object.....	203
ARM diagram .....	1853
mapping specification.....	832
Performance_rate	
application assertion .....	296
application object.....	204
ARM diagram .....	1881
mapping specification.....	905
permanent_mould_process	
mapping specification.....	900
Permanent_mould_process	
mapping specification.....	900
permanent_moulding_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1612
AIM EXPRESS short listing entity .....	1211

Permanent_moulding_process	
application assertion .....	346, 347
application object.....	204
ARM diagram.....	1885
permanent_moulding_process_type	
ARM diagram.....	1885
perpendicular_to	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1613
perpendicularity_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1613
mapping specification.....	788
person	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1614
mapping specification.....	845
Person	
application assertion .....	347, 348
application object.....	205
ARM diagram.....	1854
mapping specification.....	845
person_and_organization	
AIM diagram .....	1940
AIM EXPRESS long listing entity .....	1614
mapping specification.....	846
person_and_organization_assignment	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1614
person_and_organization_item	
AIM diagram .....	1939
AIM EXPRESS long listing type .....	1406
AIM EXPRESS short listing type .....	1009
person_and_organization_role	
AIM diagram .....	1939
AIM EXPRESS long listing entity .....	1614
Person_in_organization	
application assertion .....	297, 308, 311, 313, 332, 344, 345, 347, 348, 355, 362, 363, 374
application object.....	207
ARM diagram.....	1854
mapping specification.....	846
person_or_organization	
ARM diagram.....	1849
person_organization_select	
AIM diagram .....	1937
AIM EXPRESS long listing type .....	1407
personal_address	
AIM diagram .....	1940
© ISO 2008 — All rights reserved	2307

AIM EXPRESS long listing entity .....	1614
pin_center	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1615
AIM EXPRESS short listing entity .....	1214
Pin_center	
application assertion .....	321, 346, 348
application object.....	207
ARM diagram .....	1868
mapping specification.....	470
pin_tip	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1615
AIM EXPRESS short listing entity .....	1215
Pin_tip	
application object.....	208
Pin_tips	
application assertion .....	316, 348
ARM diagram .....	1866
mapping specification.....	471
placed_datum_target_feature	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1615
AIM EXPRESS short listing entity .....	1215
mapping specification.....	796, 802, 803, 804, 805, 806
Placed_datum_target_feature	
mapping specification.....	797
Placed_target	
application assertion .....	348
application object.....	208
ARM diagram .....	1879
mapping specification.....	796
placement	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1617
mapping specification.....	965
Planar_element	
application assertion .....	305, 315, 322, 331, 345, 348, 349, 353
application object.....	208
ARM diagram .....	1860
mapping specification.....	970
Planar_rib	
application assertion .....	348, 349
application object.....	209
ARM diagram .....	1862
mapping specification.....	477
planar_shape_representation	
AIM diagram .....	1913

AIM EXPRESS long listing entity .....	1618
mapping specification .....	970
plane	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1618
plane_angle_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1407
plane_angle_measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1618
plane_angle_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1618
plate_material_type	
ARM diagram .....	1866
plus_minus_tolerance	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1618
Plus_minus_value	
application assertion .....	340, 376
application object .....	210
ARM diagram .....	1877
mapping specification .....	797
pocket	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1618
pocket_bottom	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1623
point	
AIM diagram .....	1915
AIM EXPRESS long listing entity .....	1624
poly_loop	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1625
polyline	
AIM diagram .....	1918
AIM EXPRESS long listing entity .....	1625
position_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1625
mapping specification .....	788
positive_length_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1407
positive_plane_angle_measure	
AIM diagram .....	1942
© ISO 2008 — All rights reserved	2309

AIM EXPRESS long listing type .....	1407
pre_defined_item	
AIM diagram .....	1936
AIM EXPRESS long listing entity .....	1625
precision_qualifier	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1625
preferred_surface_curve_representation	
AIM diagram .....	1917
AIM EXPRESS long listing type .....	1407
Process activities	
units of functionality .....	24
Process plan	
units of functionality .....	26
process_or_process_relationship	
AIM diagram .....	1932
AIM EXPRESS long listing type .....	1407
process_parameter_record	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1625
AIM EXPRESS short listing entity .....	1219
mapping specification .....	943
Process_parameter_record	
application assertion .....	349
application object .....	211
ARM diagram .....	1883
mapping specification .....	943
process_pla_specification	
AIM EXPRESS short listing entity .....	1224
process_plan	
mapping specification .....	916
Process_plan	
application assertion .....	334, 344, 349, 350
application object .....	212
ARM diagram .....	1880
mapping specification .....	916
process_plan_activity	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1626
AIM EXPRESS short listing entity .....	1220
mapping specification .....	863, 870
Process_plan_security	
application object .....	213
process_plan_security	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1628
AIM EXPRESS short listing entity .....	1223
mapping specification .....	918

Process_plan_security	
application assertion .....	350
ARM diagram .....	1880
mapping specification .....	918
process_plan_select	
ARM diagram .....	1880
process_plan_specification	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1628
mapping specification .....	774
process_plan_version	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1629
AIM EXPRESS short listing entity .....	1225
mapping specification .....	914, 920
Process_plan_version	
application assertion .....	297, 350, 351, 364
application object .....	214
ARM diagram .....	1880
mapping specification .....	920
process_product_association	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1630
Process_property	
application assertion .....	324, 351, 361
application object .....	215
ARM diagram .....	1856
mapping specification .....	747
process_property_association	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1630
process_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1631
AIM EXPRESS short listing entity .....	1228
mapping specification .....	599
Process_requirement	
application assertion .....	351
application object .....	215
ARM diagram .....	1888
mapping specification .....	599
processes	
ARM diagram .....	1887
product	
AIM diagram .....	1904
AIM EXPRESS long listing entity .....	1631
AIM EXPRESS short listing imported entity modifications .....	1328
Product	
© ISO 2008 — All rights reserved	2311

definition.....	5
Product data .....	6
product_category	
AIM diagram .....	1904
AIM EXPRESS long listing entity .....	1631
product_context	
AIM EXPRESS long listing entity .....	1631
product_definition	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1631
AIM EXPRESS short listing imported entity modifications .....	1328
mapping specification.....	731, 737, 743
product_definition_context	
AIM EXPRESS long listing entity .....	1632
product_definition_effectivity	
AIM diagram .....	1906
AIM EXPRESS long listing entity .....	1632
product_definition_formation	
AIM diagram .....	1904
AIM EXPRESS long listing entity .....	1632
AIM EXPRESS short listing imported entity modifications .....	1328
mapping specification.....	853
product_definition_formation_relationship	
AIM diagram .....	1904
AIM EXPRESS long listing entity .....	1632
product_definition_process	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1632
product_definition_relationship	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1633
product_definition_shape	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1633
mapping specification.....	733, 966
mapping table .....	957
product_definition_usage	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1633
mapping specification.....	852
product_definition_with_associated_documents	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1633
product_material_composition_relationship	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1633
mapping specification.....	738
product_or_formation_or_definition	
2312	



AIM diagram .....	1934
AIM EXPRESS long listing type .....	1407
Product_quality_report	
application assertion .....	326, 351, 366
application object.....	216
ARM diagram .....	1899
mapping specification.....	765
product_related_product_category	
AIM diagram .....	1904
AIM EXPRESS long listing entity .....	1634
product_requires_version	
AIM EXPRESS long listing rule .....	1746
AIM EXPRESS short listing rules.....	1350
production_core_box	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1634
AIM EXPRESS short listing entity .....	1229
Production_core_box	
application assertion .....	300, 306, 315, 328, 351, 352, 353, 356, 358, 367
application object.....	216
ARM diagram.....	1875
mapping specification.....	485
production_die_cast_mould	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1638
AIM EXPRESS short listing entity .....	1236
mapping specification.....	636
Production_die_cast_mould	
application assertion .....	311, 315, 353, 354, 355, 368
application object.....	221
ARM diagram .....	1871
mapping specification.....	636
production_die_mould	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1640
AIM EXPRESS short listing entity .....	1240
mapping specification.....	651
Production_die_mould	
application assertion .....	312, 313, 315, 347, 355, 367
application object.....	224
ARM diagram.....	1872
mapping specification.....	651
production_investment_cast_mould	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1642
AIM EXPRESS short listing entity .....	1243
mapping specification.....	692
Production_investment_cast_mould	
© ISO 2008 — All rights reserved	2313

ISO 10303-223:2008 (E)

application assertion .....	315, 328, 355, 368
application object.....	225
ARM diagram .....	1870
mapping specification.....	692
production_pattern_definition	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1646
AIM EXPRESS short listing entity .....	1249
Production_pattern_definition	
application assertion .....	325, 340, 357, 358, 359, 366, 368
application object.....	228
ARM diagram .....	1867
mapping specification.....	507
production_rate	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1649
AIM EXPRESS short listing entity .....	1255
mapping specification.....	905
Production_rate	
application object.....	232
ARM diagram .....	1881
mapping specification.....	905
production_tool	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1642
AIM EXPRESS short listing entity .....	1244
mapping specification.....	657
Production_tool	
application assertion .....	356, 357
application object.....	225
ARM diagram .....	1870
mapping specification.....	657
production_welding	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1649
AIM EXPRESS short listing entity .....	1256
mapping specification.....	906
Production_welding	
application object.....	232
ARM diagram .....	1886
mapping specification.....	906
profile_floor	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1649
Project_order	
application assertion .....	308, 359, 360
application object.....	233
ARM diagram .....	1853

mapping specification.....	832
project_order_requires_approval	
AIM EXPRESS long listing rule.....	1746
AIM EXPRESS short listing rules.....	1351
project_order_tracking_relationships	
AIM EXPRESS long listing rule.....	1747
AIM EXPRESS short listing rules.....	1352
projected_zone_definition	
AIM diagram.....	1926
AIM EXPRESS long listing entity.....	1651
mapping specification.....	800
Projection	
application assertion.....	360, 377
application object.....	234
ARM diagram.....	1879
mapping specification.....	800
Proof_report	
application assertion.....	306
application object.....	235
ARM diagram.....	1899
mapping specification.....	770
Property.....	41, 154, 238, 330
application assertion.....	344, 360, 361, 362
application object.....	235
ARM diagram.....	1856
mapping specification.....	748
property_inspection	
mapping specification.....	944
Property_inspection	
mapping specification.....	944
Property_relationship	
application assertion.....	300
Property_bsu	
mapping specification.....	703
Property_BSU	
application assertion.....	361
application object.....	236
ARM diagram.....	1901
property_definition	
AIM diagram.....	1907
AIM EXPRESS long listing entity.....	1651
mapping specification.....	729, 747, 748, 752
property_definition_relationship	
AIM diagram.....	1907
AIM EXPRESS long listing entity.....	1652
property_definition_representation	
AIM diagram.....	1908
AIM EXPRESS long listing entity.....	1652
© ISO 2008 — All rights reserved	2315

property_inspection	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1652
AIM EXPRESS short listing entity .....	1256
Property_inspection	
application assertion .....	361
application object.....	237
ARM diagram.....	1852
property_or_shape_select	
AIM diagram .....	1927
AIM EXPRESS long listing type .....	1408
Property_parameter	
application assertion .....	319, 335, 342, 349, 350, 351, 361, 375
application object.....	237
ARM diagram.....	1858
mapping specification.....	751
property_process	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1653
property_relationship	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1653
AIM EXPRESS short listing entity .....	1257
mapping specification.....	750
Property_relationship	
application assertion .....	298, 304, 361, 369, 370
application object.....	238
ARM diagram.....	1858
mapping specification.....	750
property_select	
ARM diagram.....	1884
Property_value	
application assertion .....	330, 361
application object.....	238
ARM diagram.....	1901
mapping specification.....	704
property_value_or_relationship_select	
ARM diagram.....	1858
prototyping_or_mould_select	
ARM diagram.....	1873
protrusion	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1653
qualified_representation_item	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1654
mapping specification.....	746, 797, 809
Quality_acceptance_report	

application assertion .....	301, 362
application object.....	239
ARM diagram .....	1899
mapping specification.....	770
Quality_control	
units of functionality.....	26
quasi_uniform_curve	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1654
quasi_uniform_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1654
Radial_dimension_tolerance	
application object.....	239
ARM diagram .....	1877
mapping specification.....	801
ratio_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1408
ratio_measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1654
ratio_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1655
rational_b_spline_curve	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1655
rational_b_spline_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1655
rectangular_closed_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1655
rectangular_pattern	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1657
Red_line_part	
application assertion .....	362, 363
application object.....	239
red_lined_part	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1659
AIM EXPRESS short listing entity .....	1258
mapping specification.....	859
Red_lined_part	
ARM diagram .....	1850
mapping specification.....	859
© ISO 2008 — All rights reserved	2317

referenced_modified_datum	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1660
refractory material	
other definitions .....	8
relationship_condition	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1660
relationship_with_condition	
AIM diagram .....	1932
AIM EXPRESS long listing type .....	1408
removal_volume	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1660
replicate_feature	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1661
reporting_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1662
AIM EXPRESS short listing entity .....	1260
mapping specification .....	600
Reporting_requirement	
application assertion .....	362
application object .....	240
ARM diagram .....	1896
mapping specification .....	600
representation	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1662
mapping specification .....	704, 720, 759
representation_context	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1662
representation_item	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1662
representation_map	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1663
representation_relationship	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1663
representation_type	
ARM diagram .....	1859
represented_definition	
AIM diagram .....	1908
AIM EXPRESS long listing type .....	1408

Request_for_clarification	
application assertion .....	362, 363
application object.....	241
ARM diagram.....	1849
mapping specification.....	609
request_for_quotation	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1663
AIM EXPRESS short listing entity .....	1261
mapping specification.....	835
Request_for_quotation	
application assertion .....	308, 363
application object.....	242
ARM diagram.....	1855
mapping specification.....	835
requirement_for_action_resource	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1665
Requisition	
application assertion .....	360, 363
application object.....	244
ARM diagram.....	1853
mapping specification.....	955
Requisitions	
units of functionality.....	27
reset_or_push_pin	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1665
AIM EXPRESS short listing entity .....	1265
Reset_or_return_pins	
application assertion .....	316, 364
application object.....	245
ARM diagram.....	1866
mapping specification.....	522
Resource_acquisition_order	
application assertion .....	360
application object.....	246
ARM diagram.....	1853
mapping specification.....	841
resource_property	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1666
resource_property_representation	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1666
resource_requirement_type	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1666
© ISO 2008 — All rights reserved	2319

resource_with_material	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1666
AIM EXPRESS short listing entity .....	1266
mapping specification .....	728
Resource_with_material	
application assertion .....	364
application object .....	246
ARM diagram .....	1882
mapping specification .....	728
resource_with_representation	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1667
AIM EXPRESS short listing entity .....	1267
mapping specification .....	729
Resource_with_representation	
application assertion .....	364
application object .....	247
ARM diagram .....	1882
mapping specification .....	729
restrict_approval_status	
AIM EXPRESS long listing rule .....	1748
AIM EXPRESS short listing rules .....	1354
restrict_externally_defined_feature_definition	
AIM EXPRESS long listing rule .....	1748
AIM EXPRESS short listing rules .....	1355
restrict_name_for_known_source	
AIM EXPRESS long listing rule .....	1748
AIM EXPRESS short listing rules .....	1357
restrict_security_classification_level	
AIM EXPRESS long listing rule .....	1748
AIM EXPRESS short listing rules .....	1356
reversible_topology	
AIM diagram .....	1922
AIM EXPRESS long listing type .....	1408
reversible_topology_item	
AIM diagram .....	1922
AIM EXPRESS long listing type .....	1408
revision	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1667
AIM EXPRESS short listing entity .....	1268
revision	
mapping specification .....	923
Revision	
application assertion .....	364
application object .....	247
ARM diagram .....	1880



Revision	
mapping specification .....	923
revolved_profile	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1668
rib	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1670
AIM EXPRESS short listing entity .....	1269
mapping specification .....	477
Rib	
application object .....	248
ARM diagram .....	1862
mapping specification .....	524
rib_top	
AIM diagram .....	1903
AIM EXPRESS long listing entity .....	1672
rib_top_floor	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1673
riser	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1674
AIM EXPRESS short listing entity .....	1273
Riser	
application assertion .....	357, 359
application object .....	249
mapping specification .....	525
riser_contact	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1676
AIM EXPRESS short listing entity .....	1276
Riser_contact	
application assertion .....	357, 359, 364, 365
application object .....	248
ARM diagram .....	1865
mapping specification .....	532
riser_select	
ARM diagram .....	1865
riser_used	
ARM diagram .....	1865
Risers	
application assertion .....	320, 322, 364, 365
ARM diagram .....	1865
role_association	
AIM diagram .....	1944
AIM EXPRESS long listing entity .....	1676
role_select	
© ISO 2008 — All rights reserved	2321

AIM diagram .....	1944
AIM EXPRESS long listing type .....	1409
round_hole	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1676
rounded_end	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1678
rounded_u_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1679
roundness_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1680
roundness_tolerancemapping specification .....	788
runner .....	1680
AIM diagram .....	1910
AIM EXPRESS short listing entity .....	1277
Runner	
application assertion .....	347, 354, 374
application object.....	252
ARM diagram.....	1864
runner_or_ingate	
ARM diagram.....	1871
Runners	
application assertion .....	322, 325, 365
mapping specification.....	534
runout_zone_definition	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1682
runout_zone_orientation	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1682
runout_zone_orientation_reference_direction	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1682
sample_from	
ARM diagram.....	1851
sampled_set	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1682
AIM EXPRESS short listing entity .....	1280
mapping specification.....	945
Sampled_set	
application assertion .....	297, 299, 327, 366
application object.....	253
ARM diagram.....	1851
mapping specification.....	945

sand casting.....	8
sand_cast_design_feature	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1683
AIM EXPRESS short listing entity .....	1281
Sand_cast_design_feature	
application assertion .....	366
ARM diagram .....	1863
Sand_casting_design_feature	
application object.....	254
sand_casting_tooling	
AIM diagram .....	1905
AIM EXPRESS long listing entity .....	1683
AIM EXPRESS short listing entity .....	1282
Sand_casting_tooling	
application assertion .....	366
application object.....	255
ARM diagram .....	1867
mapping specification.....	542
sand_mould	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1684
AIM EXPRESS short listing entity .....	1284
Sand_mould	
application assertion .....	305, 321, 366, 367
application object.....	255
ARM diagram .....	1869
mapping specification.....	545
sand_mould_system	
ARM diagram .....	1869
scalar_measure	
ARM diagram .....	1893
scalar_times_vector	
AIM EXPRESS long listing FUNCTION .....	1769
second_in_minute	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1409
second_proj_axis	
AIM EXPRESS long listing FUNCTION .....	1770
secondary_tooling	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1685
AIM EXPRESS short listing entity .....	1285, 1286
Secondary_tooling	
application assertion .....	354
application object.....	257
ARM diagram .....	1871
mapping specification.....	673
© ISO 2008 — All rights reserved	2323

security_classification	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1685
security_classification_assignment	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1685
security_classification_item	
AIM diagram .....	1931
AIM EXPRESS long listing type .....	1409
AIM EXPRESS short listing type .....	1010
security_classification_level	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1686
AIM EXPRESS short listing imported entity modifications .....	1329
sequential_method	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1686
serial_action_method	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1686
set_of_reversible_topology_item	
AIM diagram .....	1922
AIM EXPRESS long listing type .....	1409
set_of_topology_reversed	
AIM EXPRESS long listing FUNCTION .....	1770
set_representation_item	
AIM diagram .....	1914
AIM EXPRESS long listing type .....	1409
Setup_activity	
application assertion .....	333
application object.....	257
ARM diagram.....	1881
mapping specification.....	906
shakeout	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1686
AIM EXPRESS short listing entity .....	1286
mapping specification.....	729
Shakeout	
application object.....	257
ARM diagram.....	1894
mapping specification.....	729
Shape representation for casting and machining	
units of functionality.....	28
shape_aspect	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1686
mapping specification.....	971, 972

Shape_aspect	
application assertion	298, 300, 307, 310, 312, 314, 323, 324, 327, 329, 337, 343, 353, 357, 361, 362, 367, 373
application object	257
ARM diagram	1859
mapping specification	971
shape_aspect_deriving_relationship	
AIM diagram	1925
AIM EXPRESS long listing entity	1686
shape_aspect_relationship	
AIM diagram	1911
AIM EXPRESS long listing entity	1687
mapping specification	818
Shape_aspect_representation	
application assertion	367
application object	258
ARM diagram	1859
mapping specification	972
shape_defining_relationship	
AIM diagram	1911
AIM EXPRESS long listing entity	1687
shape_definition	
AIM diagram	1907
AIM EXPRESS long listing type	1409
shape_definition_representation	
AIM diagram	1908
shape_dimension	
AIM diagram	1907
AIM EXPRESS long listing entity	1687
mapping specification	947
Shape_dimension	
mapping specification	947
shape_dimension_representation	
AIM diagram	1913
AIM EXPRESS long listing entity	1688
mapping specification	809
Shape_dimensions	
application assertion	367, 368, 377
application object	258
ARM diagram	1887
Shape_element	
application assertion	302, 306, 307, 309, 319, 324, 329, 330, 348, 353, 354, 357, 359, 360, 364, 367, 370, 373, 375, 377, 379
application object	259
ARM diagram	1860
mapping specification	972
shape_element_select	
ARM diagram	1860
© ISO 2008 — All rights reserved	2325

shape_representation	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1688
mapping specification .....	965, 973
Shape_representation_type	
application assertion .....	338, 341
application object .....	259
ARM diagram .....	1859
mapping specification .....	973
shape_representation_with_parameters	
AIM diagram .....	1913
AIM EXPRESS long listing entity .....	1688
Shape_representation_with_parameters	
mapping specification .....	797, 802, 803, 805, 806
mapping table .....	821
shape_select	
ARM diagram .....	1862
shell	
AIM diagram .....	1923
AIM EXPRESS long listing type .....	1409
shell_reversed	
AIM EXPRESS long listing FUNCTION .....	1771
shipping_type	
ARM diagram .....	1896
Shop_work_order	
application assertion .....	360
application object .....	260
ARM diagram .....	1853
mapping specification .....	841
shot_sleeve	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1688
AIM EXPRESS short listing entity .....	1287
mapping specification .....	674
Shot_sleeve	
application assertion .....	354, 368
application object .....	260
ARM diagram .....	1872
mapping specification .....	674
si_prefix	
AIM diagram .....	1941
AIM EXPRESS long listing type .....	1410
si_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1689
si_unit_name	
AIM diagram .....	1941
AIM EXPRESS long listing type .....	1410

simulated_property	
AIM diagram .....	1912
AIM EXPRESS long listing entity .....	1689
AIM EXPRESS short listing entity .....	1288
mapping specification .....	989
Simulated_property	
application assertion .....	373
application object .....	260
ARM diagram .....	1893
mapping specification .....	989
Simulation	
units of functionality .....	29
Simulation_exception_report	
application assertion .....	338
ARM diagram .....	1899
mapping specification .....	771
Simulation_exeption_report	
application object .....	261
simulation_input	
AIM diagram .....	1932
AIM EXPRESS long listing entity .....	1689
AIM EXPRESS short listing entity .....	1289
mapping specification .....	982
Simulation_input	
application assertion .....	368, 369, 372
application object .....	261
ARM diagram .....	1892
mapping specification .....	982
simulation_input_region	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1690
AIM EXPRESS short listing entity .....	1291
mapping specification .....	985
Simulation_input_region	
application assertion .....	369, 370
application object .....	263
ARM diagram .....	1892
mapping specification .....	985
simulation_output	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1692
AIM EXPRESS short listing entity .....	1293
mapping specification .....	987
Simulation_output	
application assertion .....	370, 372
application object .....	264
ARM diagram .....	1893
mapping specification .....	987
© ISO 2008 — All rights reserved	2327

simulation_output_region	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1692
AIM EXPRESS short listing entity .....	1294
mapping specification .....	988
Simulation_output_region	
application assertion .....	370, 371, 372
application object .....	264
ARM diagram .....	1893
mapping specification .....	988
simulation_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1693
AIM EXPRESS short listing entity .....	1296
mapping specification .....	890, 907
Simulation_process	
application assertion .....	370, 371
application object .....	265
ARM diagram .....	1891
mapping specification .....	907
Simulation_report	
application assertion .....	309
application object .....	266
ARM diagram .....	1899
mapping specification .....	771
simulation_result	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1694
AIM EXPRESS short listing entity .....	1297
mapping specification .....	990
Simulation_result	
application assertion .....	370, 371
application object .....	266
ARM diagram .....	1893
mapping specification .....	990
simulation_run	
AIM diagram .....	1927
AIM EXPRESS long listing entity .....	1695
AIM EXPRESS short listing entity .....	1298
mapping specification .....	992
Simulation_run	
application assertion .....	371, 372
application object .....	267
ARM diagram .....	1892
mapping specification .....	992
simulation_run_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1695



AIM EXPRESS short listing entity .....	1300
mapping specification .....	993
Simulation_run_relationship	
application assertion .....	372
application object .....	268
ARM diagram .....	1892
mapping specification .....	993
simulation_software	
AIM diagram .....	1933
AIM EXPRESS long listing entity .....	1696
AIM EXPRESS short listing entity .....	1300
mapping specification .....	994
Simulation_software	
application assertion .....	372
application object .....	268
ARM diagram .....	1892
mapping specification .....	994
Simulation_tooling_report	
application assertion .....	337
application object .....	269
ARM diagram .....	1899
simulation_unit	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1696
AIM EXPRESS short listing entity .....	1301
mapping specification .....	995
Simulation_unit	
application assertion .....	372, 373
application object .....	269
ARM diagram .....	1893
mapping specification .....	995
simulation_unit_state	
AIM diagram .....	1930
AIM EXPRESS long listing entity .....	1697
AIM EXPRESS short listing entity .....	1303
mapping specification .....	997
Simulation_unit_state	
application assertion .....	370, 372, 373
application object .....	270
ARM diagram .....	1893
mapping specification .....	997
single_activity_relationship	
AIM diagram .....	1929
AIM EXPRESS long listing entity .....	1697
AIM EXPRESS short listing entity .....	1304
Size_tolerance	
application assertion .....	373
application object .....	270
© ISO 2008 — All rights reserved	2329

ARM diagram .....	1877
mapping specification .....	801
slide	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1698
AIM EXPRESS short listing entity .....	1305
mapping specification .....	675
Slide	
application assertion .....	331, 354, 355, 373
application object .....	271
ARM diagram .....	1872
mapping specification .....	675
slot	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1698
slot_end	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1700
solid_model .....	8
solid_angle_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1411
solid_angle_unit	
AIM diagram .....	1941
AIM EXPRESS long listing entity .....	1702
solid_model	
AIM diagram .....	1923
AIM EXPRESS long listing entity .....	1702
source_item	
AIM diagram .....	1936
AIM EXPRESS long listing type .....	1411
special_inspection_requirement	
AIM diagram .....	1907
AIM EXPRESS long listing entity .....	1702
AIM EXPRESS short listing entity .....	1306
mapping specification .....	601
Special_inspection_requirement	
application assertion .....	327, 373
application object .....	272
ARM diagram .....	1890
mapping specification .....	601
Special_instruction	
application assertion .....	296, 351
application object .....	273
ARM diagram .....	1881
mapping specification .....	771
Specification	
application assertion .....	305, 308, 324, 326, 334, 335, 337, 361, 373, 374, 376

application object.....	274
ARM diagram.....	1900
mapping specification.....	772
Specification_usage_constraint	
application assertion.....	373
application object.....	275
ARM diagram.....	1900
mapping specification.....	774
spherical_cap	
AIM diagram.....	1903, 1909
AIM EXPRESS long listing entity.....	1703
spherical_surface	
AIM diagram.....	1920
AIM EXPRESS long listing entity.....	1704
sprue	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1704
AIM EXPRESS short listing entity.....	1307, 1313
Sprue	
application assertion.....	322, 347, 355, 365, 374
application object.....	276
ARM diagram.....	1864
sprue_and_runner_mould	
AIM diagram.....	1910
AIM EXPRESS long listing entity.....	1705
AIM EXPRESS short listing entity.....	1310
mapping specification.....	694
Sprue_and_runner_mould	
application assertion.....	329, 374
application object.....	278
ARM diagram.....	1873
mapping specification.....	694
Sprues	
mapping specification.....	549
square_u_profile	
AIM diagram.....	1911
AIM EXPRESS long listing entity.....	1706
standard_uncertainty	
AIM diagram.....	1914
AIM EXPRESS long listing entity.....	1709
stationary_die_select	
ARM diagram.....	1872
status	
ARM diagram.....	1855
Status_authority	
application assertion.....	364, 374
ARM diagram.....	1880
mapping specification.....	846
© ISO 2008 — All rights reserved	2331

step	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1709
Stop_bin	
application assertion .....	317
stop_pin	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1710
AIM EXPRESS short listing entity .....	1311
Stop_pin	
application assertion .....	375
ARM diagram .....	1866
mapping specification .....	554
straightness_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1711
mapping specification .....	788
subtype_mandatory_characterized_object	
AIM EXPRESS long listing rule .....	1749
AIM EXPRESS short listing rules .....	1357
Supplemental_document	
application assertion .....	296
application object .....	279
ARM diagram .....	1881
mapping specification .....	774
Supplier_bsu	
mapping specification .....	706
Supplier_BSU	
application assertion .....	302
application object .....	280
ARM diagram .....	1901
supported_item	
AIM diagram .....	1933
AIM EXPRESS long listing type .....	1411
surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1711
surface_curve	
AIM diagram .....	1917
AIM EXPRESS long listing entity .....	1711
surface_finish	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1711
AIM EXPRESS short listing entity .....	1312
mapping specification .....	909
Surface_finish	
application object .....	280
ARM diagram .....	1886

mapping specification.....	909
surface_of_linear_extrusion	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1711
surface_of_revolution	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1711
surface_profile_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1712
mapping specification.....	788
Surface_property	
application assertion .....	361, 375
ARM diagram.....	1856
mapping specification.....	752
Surface_propertyt	
application object.....	280
surface_weights_positive	
AIM EXPRESS long listing FUNCTION .....	1771
swept_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1712
symmetric_shape_aspect	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1712
symmetry_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1712
mapping specification.....	788
tangent	
AIM diagram .....	1925
AIM EXPRESS long listing entity .....	1712
taper	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1713
Target_area	
application assertion .....	375
application object.....	281
ARM diagram.....	1879
mapping specification.....	801
Target_circle	
application object.....	281
ARM diagram.....	1879
mapping specification.....	802
Target_line	
application object.....	281
ARM diagram.....	1879
mapping specification.....	803
© ISO 2008 — All rights reserved	2333

Target_point	
application object.....	282
ARM diagram.....	1879
mapping specification.....	804
Target_rectangle	
application object.....	282
ARM diagram.....	1879
mapping specification.....	804
tee_profile	
AIM diagram.....	1911
AIM EXPRESS long listing entity.....	1714
tensor_measure	
ARM diagram.....	1893
text	
AIM diagram.....	1945
AIM EXPRESS long listing type.....	1411
thermodynamic_temperature_measure	
AIM diagram.....	1942
AIM EXPRESS long listing type.....	1411
Thickness_tolerance	
application assertion.....	375, 807
application object.....	282
ARM diagram.....	1877
thread	
AIM diagram.....	1903, 1909
AIM EXPRESS long listing entity.....	1717
time_assignment	
AIM diagram.....	1939
AIM EXPRESS long listing entity.....	1721
time_item	
AIM diagram.....	1939
AIM EXPRESS long listing type.....	1411
AIM EXPRESS short listing type.....	1010
time_measure	
AIM diagram.....	1942
AIM EXPRESS long listing type.....	1411
time_measure_with_unit	
AIM diagram.....	1942
AIM EXPRESS long listing entity.....	1721
time_role	
AIM diagram.....	1939
AIM EXPRESS long listing entity.....	1721
time_unit	
AIM diagram.....	1941
AIM EXPRESS long listing entity.....	1721
tolerance_inspection	
mapping specification.....	951
Tolerance_inspection	

mapping specification.....	951
tolerance_definition_select	
ARM diagram.....	1877
tolerance_inspection	
AIM diagram.....	1907
AIM EXPRESS long listing entity.....	1722
Tolerance_inspection	
application assertion.....	375
application object.....	283
ARM diagram.....	1852
Tolerance_limit	
application assertion.....	340, 376
application object.....	283
ARM diagram.....	1877
mapping specification.....	807
tolerance_method_definition	
AIM diagram.....	1926
AIM EXPRESS long listing type.....	1411
Tolerance_placement	
application assertion.....	313
application object.....	284
ARM diagram.....	1878
mapping table.....	808
Tolerance_range	
application assertion.....	341, 376
application object.....	284
ARM diagram.....	1877
mapping specification.....	809
tolerance_requirement	
AIM diagram.....	1907
AIM EXPRESS long listing entity.....	1722
AIM EXPRESS short listing entity.....	1313
mapping specification.....	603
Tolerance_requirement	
application assertion.....	301, 375, 376, 377
application object.....	285
ARM diagram.....	1896
mapping specification.....	603
tolerance_select	
AIM diagram.....	1924
AIM EXPRESS long listing type.....	1412
tolerance_type	
ARM diagram.....	1896
tolerance_value	
AIM diagram.....	1926
AIM EXPRESS long listing entity.....	1723
mapping specification.....	797
Tolerance_value	
© ISO 2008 — All rights reserved	2335

ISO 10303-223:2008 (E)

application assertion .....	313, 376
application object.....	287
ARM diagram .....	1877
mapping specification.....	811
tolerance_zone	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1724
mapping specification.....	813
Tolerance_zone	
application assertion .....	323, 376, 377
application object.....	287
ARM diagram .....	1879
mapping specification.....	813
tolerance_zone_definition	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1724
mapping specification.....	813
Tolerance_zone_definition	
application assertion .....	376, 377
application object.....	288
ARM diagram .....	1879
mapping specification.....	813
tolerance_zone_form	
AIM diagram .....	1926
AIM EXPRESS long listing entity .....	1724
tool_location_select	
ARM diagram .....	1871
tool_verification	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1724
AIM EXPRESS short listing entity .....	1315
mapping specification.....	952
Tool_verification	
application assertion .....	377
application object.....	288, 290
ARM diagram .....	1887
mapping specification.....	952
Tooling	
application assertion .....	309, 342, 371, 377, 378
ARM diagram .....	1850
mapping specification.....	862
tooling_material	
ARM diagram .....	1857
tooling_process	
AIM diagram .....	1928
AIM EXPRESS long listing entity .....	1725
AIM EXPRESS short listing entity .....	1317
mapping specification.....	910



Tooling_process	
application assertion .....	378
application object.....	290
ARM diagram .....	1887
mapping specification.....	910
Tooling_report	
mapping specification.....	775
tooling_select	
ARM diagram.....	1887
Tooling_verification	
application assertion .....	378
topological_representation_item	
AIM diagram .....	1915
AIM EXPRESS long listing entity .....	1727
mapping specification.....	964
topology_reversed	
AIM EXPRESS long listing FUNCTION.....	1772
toroidal_surface	
AIM diagram .....	1920
AIM EXPRESS long listing entity .....	1727
total_runout_tolerance	
AIM diagram .....	1924
AIM EXPRESS long listing entity .....	1727
mapping specification.....	788
transformation	
AIM diagram .....	1916
AIM EXPRESS long listing type .....	1412
transition_code	
AIM diagram .....	1918
AIM EXPRESS long listing type .....	1412
transition_feature	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1727
AIM EXPRESS short listing imported entity modifications.....	1329
mapping specification.....	818, 819, 822
Transition_feature	
application assertion .....	304
application object.....	292
ARM diagram.....	1861
mapping specification.....	816, 822
transition_feature_life_cycle	
AIM EXPRESS long listing rule.....	1749
Transition_feature_life_cycle	
AIM EXPRESS short listing rules.....	1358
transition_feature_on_part_boundary	
AIM EXPRESS long listing rule.....	1749
Transition_feature_on_part_boundary	
AIM EXPRESS short listing rules.....	1358
© ISO 2008 — All rights reserved	2337

trimming_select	
AIM diagram .....	1915
AIM EXPRESS long listing type .....	1412
turned_knurl	
AIM diagram .....	1903, 1909
AIM EXPRESS long listing entity .....	1728
type_of_output	
ARM diagram .....	1881
type_qualifier	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1731
mapping specification .....	807
type_used	
ARM diagram .....	1865
uncertainty_measure_with_unit	
AIM diagram .....	1942
AIM EXPRESS long listing entity .....	1731
uncertainty_qualifier	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1731
uniform_curve	
AIM diagram .....	1919
AIM EXPRESS long listing entity .....	1731
uniform_surface	
AIM diagram .....	1921
AIM EXPRESS long listing entity .....	1731
unit	
AIM diagram .....	1941
AIM EXPRESS long listing type .....	1412
Unit of functionality .....	6
UoF .....	9
using_items	
AIM EXPRESS long listing FUNCTION .....	1772
using_representations	
AIM EXPRESS long listing FUNCTION .....	1773
v	
mapping specification .....	788
valid_calendar_date	
AIM EXPRESS long listing FUNCTION .....	1773
valid_measure_value	
AIM EXPRESS long listing FUNCTION .....	1774
valid_time	
AIM EXPRESS long listing FUNCTION .....	1774
valid_units	
AIM EXPRESS long listing FUNCTION .....	1775
value_qualifier	
AIM diagram .....	1914
AIM EXPRESS long listing type .....	1412

value_range	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1731
AIM EXPRESS short listing entity .....	1320
mapping specification .....	809
value_representation_item	
AIM diagram .....	1914
AIM EXPRESS long listing entity .....	1732
value_select	
ARM diagram .....	1893
vector	
AIM diagram .....	1916
AIM EXPRESS long listing entity .....	1732
vector_difference	
AIM EXPRESS long listing FUNCTION .....	1777
vector_measure	
ARM diagram .....	1893
vector_or_direction	
AIM diagram .....	1916
AIM EXPRESS long listing type .....	1412
vee_profile	
AIM diagram .....	1911
AIM EXPRESS long listing entity .....	1732
vent	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1735
AIM EXPRESS short listing entity .....	1322
Vent	
application assertion .....	305, 357, 365, 378
application object .....	292
ARM diagram .....	1865
mapping specification .....	555
vent_material	
ARM diagram .....	1895
vent_type	
ARM diagram .....	1865
versioned_action_request	
AIM diagram .....	1931
AIM EXPRESS long listing entity .....	1735
mapping specification .....	606, 609
vertex	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1735
vertex_loop	
AIM diagram .....	1922
AIM EXPRESS long listing entity .....	1736
vertex_point	
AIM diagram .....	1922
© ISO 2008 — All rights reserved	2339

AIM EXPRESS long listing entity .....	1736
view_reference	
AIM diagram .....	1934
AIM EXPRESS long listing entity .....	1736
AIM EXPRESS short listing entity .....	1323
mapping specification .....	775
View_reference	
application assertion .....	297, 324, 379
application object .....	293
ARM diagram .....	1859
mapping specification .....	775
visual_appearance	
ARM diagram .....	1856
volume_measure	
AIM diagram .....	1942
AIM EXPRESS long listing type .....	1413
Wax_or_EPS_pattern_select	
ARM diagram .....	1873
wear	
ARM diagram .....	1898
weight	
ARM diagram .....	1856
well	
AIM diagram .....	1910
AIM EXPRESS long listing entity .....	1736
AIM EXPRESS short listing entity .....	1324
Well	
application assertion .....	365, 374, 379
application object .....	294
ARM diagram .....	1865
mapping specification .....	558
Width_dimension	
application assertion .....	379, 814
application object .....	295
ARM diagram .....	1877
year_number	
AIM diagram .....	1938
AIM EXPRESS long listing type .....	1413

-----

---

---

**ICS 25.040.40**

Price based on 379 pages