INTERNATIONAL STANDARD

**ISO 10303-240**

First edition
2005-12-15

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

## Part 240: Application protocol: Process plans for machined products

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

*Partie 240: Protocole d'application: Plan des processus pour produits usinés*

Reference number
ISO 10303-240:2005(E)

**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.



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
Web www.iso.org

Published in Switzerland

# Contents

Page

**Figures**



 

 

**Tables**



 

# 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 10303-240 was prepared by Technical Committee ISO TC184/SC4, *Industrial automation systems and integration*, Subcommittee *SC4 Industrial data*.

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

Each part of this International Standard 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 is a member of the 200 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 exchange, archiving and sharing of computer-interpretable process plans for numerical control and manually machined parts. The intent of this AP is to allow for the exchange, archiving and sharing of data between dissimilar Computer-Aided Process Planning (CAPP) systems.

A process plan is a set of instructions. These instructions are used by programmers to generate machine tool controller programs which, when executed, drive the tool motion to remove material. Process planning takes place after the preliminary engineering design is completed but before the manufacturing process has begun, and the process plan effort requires feedback for design completeness. Design information is communicated to a process planner who identifies the manufacturing steps necessary to transform the material specified by the design into a product.

This AP specifies the data contained within a process plan as opposed to the data necessary to perform process planning functions. Included in this AP are the relationships that exist between the different process plan data items as well as the relationships that exist between these data items and the product definition data. Product definition data includes data items from design process such as geometry, surface finish and machining tolerance.

This application protocol defines the context, scope, and information requirements for digitally represented process plans for numerical control and manually machined parts and specifies the integrated resources necessary to satisfy these requirements.

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 the application protocol and summarizes the functionality and data covered by the AP. 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 may 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. A graphical representation of the AIM is given in Annex H. Additional requirements for specific implementation methods are given in Annex C.

 

Figure 1 contains the data planning model that provides a high-level description of the requirements for this application protocol, as well as identifying that some relationship exists between the basic data components.

The planning model illustrates that an process plan is composed of one or more sequenced activities. The activities can be broken into several closely related sub-activities. Activities are associated with the product definition data and will be used in the production of the product shape data. Each activity describes in detail the associated processing and the type and amount of resource required with additional special instructions, if necessary. An process plan also includes administrative data pertinent to the management of the process plan.

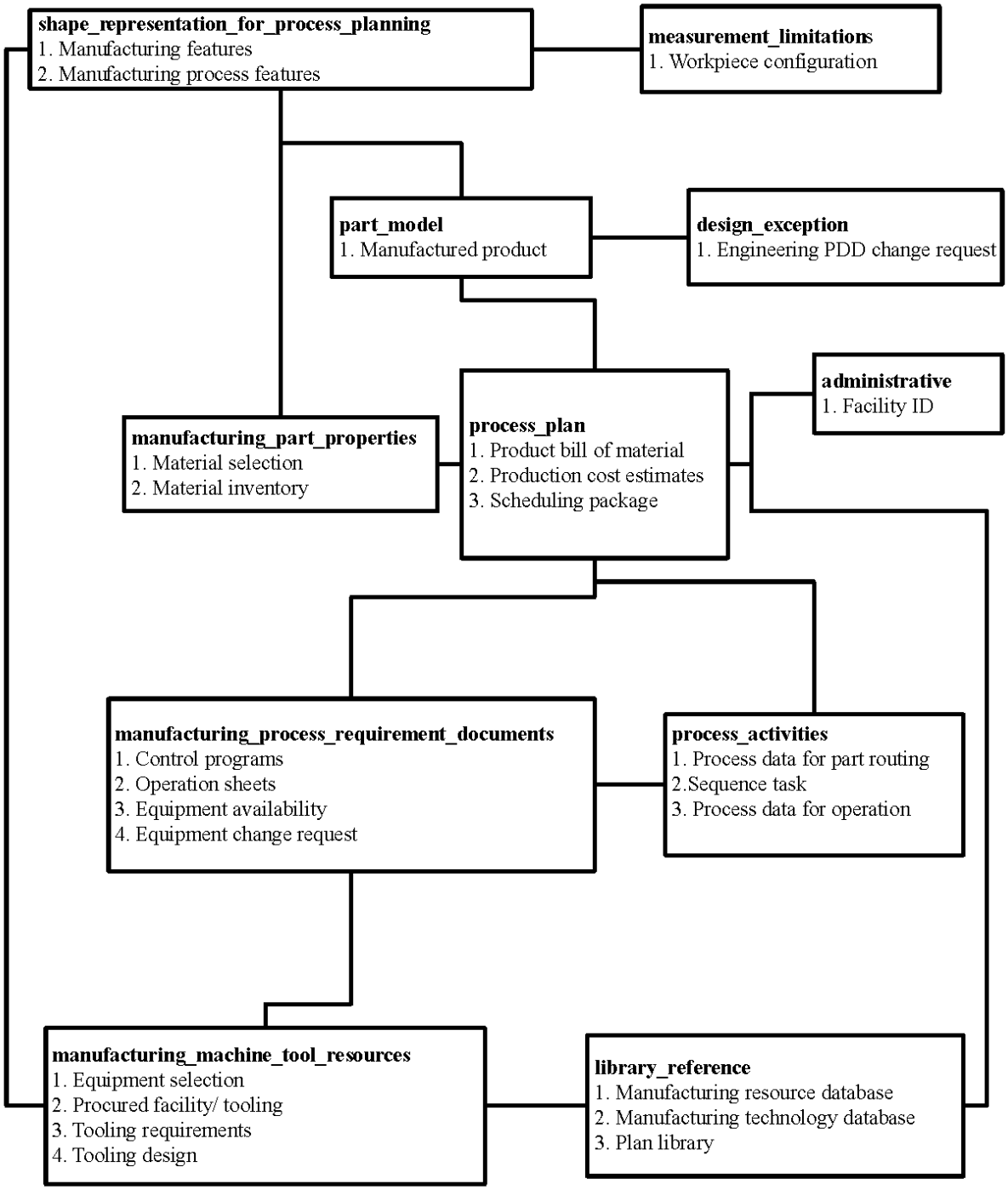


**Figure 1 — Process planning data planning model**

 

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

# Part 240: Application protocol: Process plans for machined products

## 1 Scope

This part of ISO 10303 specifies the use of the integrated resources necessary for the scope and information requirements for the exchange, sharing and long term data retention of computer-interpretable process plan information for both numerical control (NC) and manually operated applications, and associated product definition data.

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:

— information out of the planning activity that is contained in the process plans for machined parts which includes:

  — numerical controlled machines;

  — manual operations.

— the manufacture of a single piece mechanical part, and assemblies of single piece parts for manufacturing purpose which includes:

  — process data for part routing which includes manufacturing process and setup sequencing;

  — process data for operation.

— interface for capturing technical data out of the upstream application protocols which includes:

  — product definition data, including tracking a design exception notice of a part;

  — initial material definition data.

— technical data for and/or out of the process planning for machined pats which includes:

  — machining features for defining shapes necessary for manufacturing;

  — machining feature classification structure;

  — geometric and dimensional tolerances of the parts being manufactured;

 

  - materials, and properties of the parts being manufactured.

- references to standards and specifications declared in the process plan;
- work instructions for the tasks required to manufacture a part, using 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.
- information required to support NC programming of processes specified in the process plan;

NOTE 2 This includes product definition, administrative data, machine, tooling, and material requirements.

- information required to support in-process inspection specified in the process plan;

NOTE 3 In-process inspection includes such tasks as using gauge blocks or performing a probing operation to verify the dimensional constraints placed upon the part.

- shop floor information specified in the process plan;
- information for production planning specified in the process plan;

NOTE 4 Information for production planning contains items such as process data for part routing and required resource to perform the planned activity.

NOTE 5 Shop floor information contains such items as part routing, machine setup, and part loading instructions.

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

- NC program, source programs, and specific machine tool controller codes;
- NC process information derived from, or required for, manufacturing preplanning activities;

NOTE 6 This includes information from activities such as factory capacity planning, scheduling, producibility analysis, and statistical process control.

- continuous processes;

NOTE 7 Continuous process is the control of a process that requires feedback to determine new parameters such as those in adaptive control, real-time inspection and automatic corrections and in the manufacture of chemical and plating products.

- inspection processes that require an inspection plan;

NOTE 8 Inspection processes refer to inspection that occurs outside the context of the NC machining process, such as removing the part and remounting it on a Coordinate Measuring Machine (CMM).

- drawing and production illustration contents;

 

— make or buy analysis activities;

— actual costing data;

— production control and scheduling analysis;

— production planning functions;

— actual execution of the process plan or associated NC programs.

## 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 8824-1, *Information Technology — Abstract Syntax Notation One (ASN.1) — Specification of Basic Notation — Part 1.*

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

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

ISO 10303-21, *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 resources: Fundamentals of product description and support.*

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

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

ISO 10303-44, *Industrial automation systems and integration — Product data representation and exchange — Part 44: Integrated generic resources: Product structure configuration.*

ISO 10303-45, *Industrial automation systems and integration — Product data representation and exchange — Part 45: Integrated generic resources: Materials.*

ISO 10303-47, *Industrial automation systems and integration — Product data representation and exchange — Part 47: Integrated generic resources: 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 10303-501, *Industrial automation systems and integration — Product data representation and exchange — Part 501: Application interpreted construct: Edge-based wireframe.*

ISO 10303-502, *Industrial automation systems and integration — Product data representation and exchange — Part 502: Application interpreted construct: Shell-based wireframe.*

ISO 10303-507, *Industrial automation systems and integration — Product data representation and exchange — Part 507: Application interpreted construct: Geometrically bounded surface.*

ISO 10303-509, *Industrial automation systems and integration — Product data representation and exchange — Part 509: Application interpreted construct: Manifold surface.*

ISO 10303-510, *Industrial automation systems and integration — Product data representation and exchange — Part 510: Application interpreted construct: Geometrically bounded wireframe.*

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

ISO 10303-512, *Industrial automation systems and integration — Product data representation and exchange — Part 512: Application interpreted construct: Faceted boundary representation.*

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.*

ISO 13584-26, *Industrial automation systems and integration — Parts library — Part 26: Logical resource: Information supplier identification.*

ISO 13584-42, *Industrial automation systems and integration — Parts library — Part 42: Description methodology: Methodology for structuring part families.*

IEC/ISO Directives, Part 2, *Rules for the structure and drafting of International Standards*, Fourth edition, 2001.

 

# 3 Terms, definitions and abbreviations

## 3.1 Terms defined in ISO 10303-1

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-1 apply.

— abstract test suite (ATS);

— application;

— application activity model (AAM);

— application context;

— application interpreted model (AIM);

— application object;

— application protocol (AP);

— application reference model (ARM);

— computer aided design (CAD);

— computer aided manufacture (CAM);

— conformance class;

— conformance requirement;

— data;

— data exchange;

— generic resource;

— information;

— integrated generic resource;

— integrated resource;

— interpretation;

— model;

— PICS proforma;

— product;

— product data;

— product information model;

— resource construct;

— test purpose;

— unit of functionality (UoF);

— validation.

## 3.2 Terms defined in ISO 10303-21

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-21 apply.

— implementation method.

## 3.3 Terms defined in ISO 10303-31

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-31 apply.

— conformance class;

— conformance testing;

— postprocessor;

— preprocessor.

## 3.4 Other definitions

For the purposes of this part of ISO 10303, the following definitions apply:

**3.4.1**
**computer-aided process planning (CAPP) system**
a commercial or proprietary software application used to assist process planners in making a process plan.

**3.4.2**
**direct numerical control (DNC) system**
a computer system that transmits machine-tool controller codes directly from a host computer to the machine-tool controller. The machine-tool controller transmits back its status to the host system.

**3.4.3**
**engineering bill of materials (EBOM)**
the list of part numbers and assemblies that make up the design engineering configuration that contains the raw stock size and the material specification.

 

**3.4.4**
**engineering change notice (ECN)**
a document released by design engineering to production that informs production of pending changes to a specific design.

**3.4.5**
**engineering change order (ECO)**
an authorization to implement changes to a specific engineering design.

**3.4.6**
**fixture coordinate system**
a coordinate system on a fixture for the location and orientation of the fixture in a machine tool coordinate system.

**3.4.7**
**group technology (GT)**
a manufacturing philosophy in which similar parts are identified and grouped together to take advantage of their similarities in manufacturing and design.

**3.4.8**
**inventory item**
a tangible object that must be accounted for on an inventory manifest.

**3.4.9**
**machine coordinate system**
three mutually perpendicular planes which define the axis and origin for a machine location.

**3.4.10**
**machined part (MP)**
a product shaped by removing material using one or more machine tools.

**3.4.11**
**manufacturing bill of materials**
a list of materials required for the machining process plan.

EXAMPLE A product has an engineering bill of materials of: left hinge, right holder, bolt, washer, locknut, bearing washer. The left hinge, right holder, and bearing washer need to have a hole machined as a loose assembly, they are then separated. The bearing washer is returned to stores and the left hinge, right holder have other machining work done prior to being sent off to another site for heat treatment and plating. The engineering and product build will show six items but our machining process plan needs a separate bill of materials with just left hinge, right holder, and bearing washer.

**3.4.12**
**material resource planning (MRP) system**
a software application that controls and schedules the flow of materials as required by production.

**3.4.13**
**numerical control (NC)**
a machine tool controlled by a programmable computerized system.

**3.4.14**
**part coordinate system**
three mutually perpendicular planes which define the axis and origin for a part location.

**3.4.15**
**part program**
a set of instructions used to transform the part specification from an engineering drawing or 3D model to a sequence of processing steps to be performed on an NC machine.

**3.4.16**
**process plan (PP)**
the detailed method for the machining of a part. It includes a sequence of steps to be executed that call out the selected machines, fabrication, operation steps, setups, routings, and in-process inspection requirements for the manufacturing of a machined part.

**3.4.17**
**product definition data (PDD)**
the data that pertains to the design description

## 3.5 Abbreviations

For purposes of this part of ISO 10303, the following abbreviations apply:

| | |
|---|---|
| AAM | application activity model |
| AIC | application interpreted construct |
| AIM | application interpreted model |
| AP | application protocol |
| ARM | application reference model |
| ATS | abstract test suite |
| B-rep | boundary representation |
| CAD | computer aided design |
| CAPP | computer-aided process planning |
| DBMS | database management system |
| DNC | direct numerical control |
| EBOM | engineering bill of materials |
| ECN | engineering change notice |
| ECO | engineering change order |

 

| | |
|---|---|
| GT | group technology |
| ID | identification |
| IDEF0 | ICAM definition language 0 |
| IDEF1X | ICAM definition language 1 - extended |
| MP | machined part |
| MRP | material resource planning |
| NC | numerical control |
| PDD | product definition data |
| PICS | protocol implementation conformance statement |
| PP | process plan |
| UoF | units of functionality |

# 4 Information requirements

This clause specifies the information required for manual or numerical control process plans for machined parts.

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 application protocol.

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 within the scope of this application protocol in Annex F.

NOTE 3 The mapping table specified in 5.1 shows how the integrated resources and application interpreted constructs are used to meet the information requirements of this application protocol.

## 4.1 Units of functionality

This subclause specifies the units of functionality for the process planning application protocol. This part of ISO 10303 specifies the following units of functionality:

— administrative;

— design_exception;

— library_reference;

— manufacturing_machine_tool_resources;

— manufacturing_process_requirement_documents;

— manufacturing_part_properties;

— measurement_limitations;

— part_model;

— process_activities;

— process_plan;

— shape_representation_for_process_planning.

The units of functionality and a description of the functions that each UoF supports are given below. The application objects included in the UoFs are defined in 4.2.

### 4.1.1 administrative

The administrative UoF contains the information required for the management of process plan. This information contains sufficient detail to allow for the identification, release and revision of the process plan.

The following application objects are used by the administrative UoF:

— Company;

— Company_contract_assignment;

— Organization;

— Planning_group_member;

— Status_authority.

 

### 4.1.2 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 process planning.

The following application objects are used by the design_exception UoF:

— Design_exception_notice;

— Engineering_change_order;

— Engineering_change_proposal.

### 4.1.3 library_reference

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

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

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.

### 4.1.4 manufacturing_machine_tool_resources

The manufacturing_machine_tool_resources UoF contains the definitions of tools, fixtures, and jigs used to accomplish the fabrication of a part.

The following application objects are used by the manufacturing_machine_tool_resources UoF:

— Controller;

— Fixture_assembly;

— Fixture_assembly_element;

— Generic_manufacturing_resource;

— In_facility_location;

— Machine;

— Pallet;

— Part_holding_position;

— Resource_with_material;

— Resource_with_representation;

— Tool_assembly;

— Tool_assembly_element;

— Tool_body;

— Tool_magazine_turret_carousel;

— Work_cell;

— Workstation.

### 4.1.5 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.

The following application objects are used by the manufacturing_part_properties UoF:

— Alternate_material;

— Descriptive_parameter;

— Hardness;

— Material;

— Material_property;

— Numeric_parameter;

— Numeric_parameter_with_tolerance;

— Part_property;

 

— Process_property;

— Property;

— Property_parameter;

— Surface_property.

### 4.1.6 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.

The following application objects are used by the manufacturing_process_requirement_documents UoF:

— Contract;

— Controller_program;

— Digital_file;

— Document_assignment;

— Document_file;

— Document_file_properties;

— Document_file_relationship;

— Design_reference;

— Executable;

— External_file_identification;

— External_schema_definition;

— Fixture_contract;

— Hardcopy;

— Illustration;

— Material_specification;

— Partial_document_assignment;

— Part_dimensioning_standard;

— Special_capability;

— Special_instruction;

— Specification;

— Supplemental_document;

— Tool_contract;

— Tool_placement_instruction;

— View_reference.

### 4.1.7 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.

The following application objects are used by the measurement_limitations UoF:

— Angular_dimension_tolerance;

— Angular_size_dimension_tolerance;

— Curved_dimension_tolerance;

— Diameter_dimension_tolerance;

— Dimensional_tolerance;

— Distance_along_curve_tolerance;

— Externally_defined_size_dimension;

— Height_dimension;

— Geometric_tolerance;

— Length_dimension;

— Limits_and_fits;

— Location_dimension_tolerance;

— Location_tolerance;

 

— Machining_tolerance;

— Plus_minus_value;

— Radial_dimension_tolerance;

— Size_tolerance;

— Thickness_tolerance;

— Tolerance_limit;

— Tolerance_range;

— Tolerance_value.

— Width_dimension.

### 4.1.8 part_model

The part_model UoF contains the information necessary to identify the part that is to be input to the process planning 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 objects are used by the part_model UoF:

— Manufactured_assembly;

— Manufactured_assembly_relationship;

— Mating_definition;

— Mating_definition_relationship;

— Mating_relationship;

— Part_version;

— Single_piece_part.

### 4.1.9 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.

The following application objects are used by the process_activities UoF:

— Activity;

— Allowed_time;

— Alternate_activity;

— Ancillary_activity;

— Ancillary_setup;

— Continuous_process;

— Fixture_machine_mounting;

— Fixture_machine_unmounting;

— Fixture_pallet_mounting;

— Fixture_pallet_unmounting;

— Fixture_setup;

— Machine_parameters;

— Machine_setup;

— Machine_usage;

— Machining_process;

— Manufacturing_activity;

— Manufacturing_process;

— Mounting_position;

— Non_machining_process;

— Pallet_machine_mounting;

— Pallet_machine_unmounting;

— Part_fixture_mounting;

— Part_fixture_unmounting;

— Part_machine_mounting;

 

— Part_machine_unmounting;

— Part_mounting;

— Part_routing;

— Part_unmounting;

— Performance_rate;

— Process_activity;

— Production_rate;

— Setup_activity;

— Tool_setup;

— Validation.

### 4.1.10 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.

The following application objects are used by the process_plan UoF:

— Alternate_process_plan;

— Feature_dependency;

— Feature_identification_item;

— Feature_interaction;

— Feature_process;

— Process_plan_security;

— Process_plan_version;

— Range_of_parts;

— Revision.

### 4.1.11 shape_representation_for_process_planning

The shape_representation_for_process_planning 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.

The following application objects are used by the shape_representation_for_process_planning UoF:

— Base_shape;

— Block_base_shape;

— Cylindrical_base_shape;

— Explicit_base_shape_representation;

— Geometric_model;

— Implicit_base_shape_representation;

— Intermediate_shape;

— Manufacturing_process_feature;

— Model_element;

— Ngon_base_shape;

— Object_element_shape_representation;

— Orientation;

— Part_placement;

— Part_shape;

— Shape_aspect;

— Shape_aspect_representation;

— Shape_representation_type.

## 4.2 Application objects

This subclause specifies the application objects for them mechanical product definition for process planning using machining features 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 Manufacturing_activity (see 4.2.76), Setup_activity (see 4.2.128), or Alternate_-activity (see 4.2.3). The data associated with an Activity are the following:

— activity_information;

— activity_number;

— constrained_by;

— description;

— duration;

— frequency;

— graphics_representation;

— name;

— organization_id;

— references;

— resources.

#### 4.2.1.1 activity_information

The activity_information specifies the Special_instruction (see 4.2.135) for the purpose of planning a machining activity. The activity_information need not be specified for a particular Special_instruction. There may be more than one activity_information for a Special_instruction. See 4.3.6 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 constrained_by

The constrained_by specifies Supplemental_document (see 4.2.138) that is the additional documentation that may be referenced to further define the activity. The constrained_by need not be specified for a particular Activity. See 4.3.7 for the application assertion.

### 4.2.1.4 description

The description specifies the word, or group of words, that provide either instructions for the Activity to be performed or state the objective to be achieved by the Activity.

EXAMPLE These descriptions might contain:
— drill and ream all four 3" diameter through holes;
— fabricate hub mounting holes;
— fixture the shaft per diagram “A1234" and machine per NC Program “XYZ”.

### 4.2.1.5 duration

The duration specifies the Performance_rate (see 4.2.111)which is the amount of time necessary to complete the task. See 4.3.5 for the application assertion.

### 4.2.1.6 frequency

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

### 4.2.1.7 graphics_representation

The graphics_representation specifies the visual representation of information necessary for the actrivity. There may be more than one graphics_representation for a Design_reference (See 4.2.24), Illustration (see 4.2.59), or View_reference (see 4.2.153). The graphics_representation need not be specified for a particular Activity. See 4.3.1, 4.3.3, and 4.3.8 for the application assertion.

### 4.2.1.8 name

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

### 4.2.1.9 organization_id

The organization_id specifies the unique identifier for the organization responsible for performing the Activity.

### 4.2.1.10 references

The references specifies the Material (see 4.2.79) of the Part_version (see 4.2.109) necessary to perform the activity. References need not be specified for a particular Activity. See 4.3.4 for the application assertion.

### 4.2.1.11 resources

The resources specifies the Generic_manufacturing_resource (see 4.2.53) that defines the additional resources that may be necessary to perform the activity. The resources need not be specified for a particular Activity (see 4.2.79). See 4.3.2 for the application assertion.

 

### 4.2.2 Allowed_time

An Allowed_time is the total time allocated for the completion of an activity. An Allowed_time is a type of Performance_rate (see 4.2.111).

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;

— standard_time.

#### 4.2.2.1 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.

#### 4.2.2.2 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.

### 4.2.3 Alternate_activity

An Alternate_activity is the identification of a separate activity that can be used to replace the current activity within the same process plan. The data associated with an Alternate_activity is the following:

— primary_activity.

#### 4.2.3.1 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 4.3.9 for the application assertion.

### 4.2.4 Alternate_material

An Alternate_material is the identification of a secondary material that may be used when the primary material choice is not available. The data associated with an Alternate_material are the following:

EXAMPLE Figure 14 illustrates a note block for a part with Alternate_material.

NOTE The Alternate_material definition is derived from clause 4.2.1 of ISO 10303-224.

— alternate_ranking;

— material_substitute.

#### 4.2.4.1 alternate_ranking

The alternate_ranking specifies the order for selecting Alternate_material objects in the event the primary Material is not available.

#### 4.2.4.2 material_substitute

The material_substitute specifies the material to be used as the substitute. See 4.3.10 for the application assertion.

### 4.2.5 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 are the following:

— alternate_ranking;

— plan_substitute.

#### 4.2.5.1 alternate_ranking

The alternate_ranking specifies the order for selecting Alternate_process_plan objects in the event the primary Process_plan_version (see 4.2.116) is not available.

#### 4.2.5.2 plan_substitute

The plan_substitute specifies Process_plan_version (see 4.2.116) that is to be used as a substitute process plan. See 4.3.11 for the application assertion.

 

### 4.2.6 Ancillary_activity

An Ancillary_setup is a type of Manufacturing_activities (see 4.2.76) that defines an ancillary action. The data associated with an Ancillary_activity are the following:

— method.

#### 4.2.6.1 method

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

### 4.2.7 Ancillary_setup

An Ancillary_setup is a type of Setup_activity (see 4.2.128) that defines the setup for a an ancillary action Activity (see 4.2.1). The data associated with an Ancillary_setup are the following:

— setup_type.

#### 4.2.7.1 setup_type

A setup_type specifies the descriptive placement and location information for an ancillary activity.

### 4.2.8 Angular_dimension_tolerance

An Angular_dimension_tolerance is a type of Location_tolerance (see 4.2.67) 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 The Angular_dimension_tolerance definition is derived from clause 4.2.3 of ISO 10303-224.

NOTE 2 Figure 2 illustrates the Angular_dimension_tolerance.

The data associated with an Angular_dimension_tolerance are the following:

— plane_and_direction.

#### 4.2.8.1 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 4.3.12 for the application assertion.

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

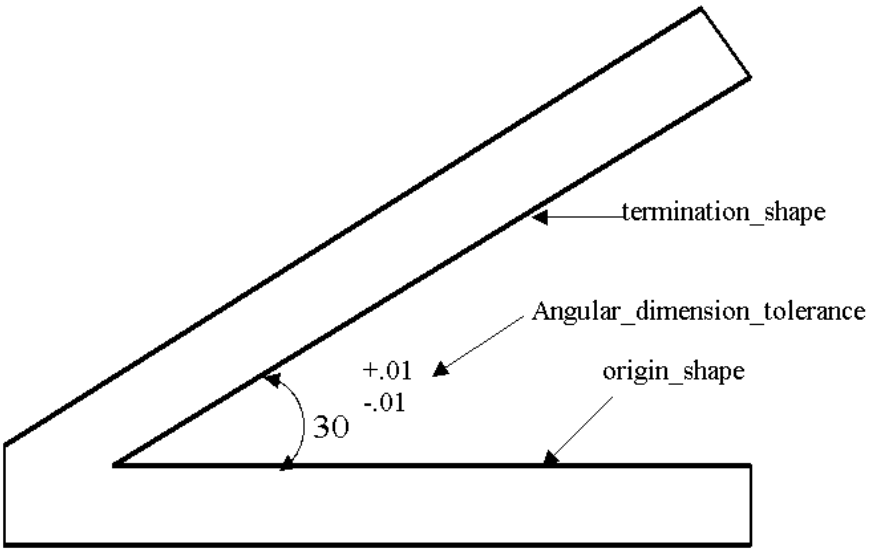


**Figure 2 — Angular_dimension_tolerance**

## 4.2.9 Angular_size_dimension_tolerance

An Angular_size_dimension_tolerance is a type of Size_tolerance (see 4.2.133) that specifies the allowable variation on the size or gap formed by two angular elements of the shape of a part.

NOTE 1 The Angular_size_dimension_tolerance definition is derived from clause 4.2.4 of ISO 10303-224.

NOTE 2 Figure 3 illustrates the Angular_size_dimension_tolerance.

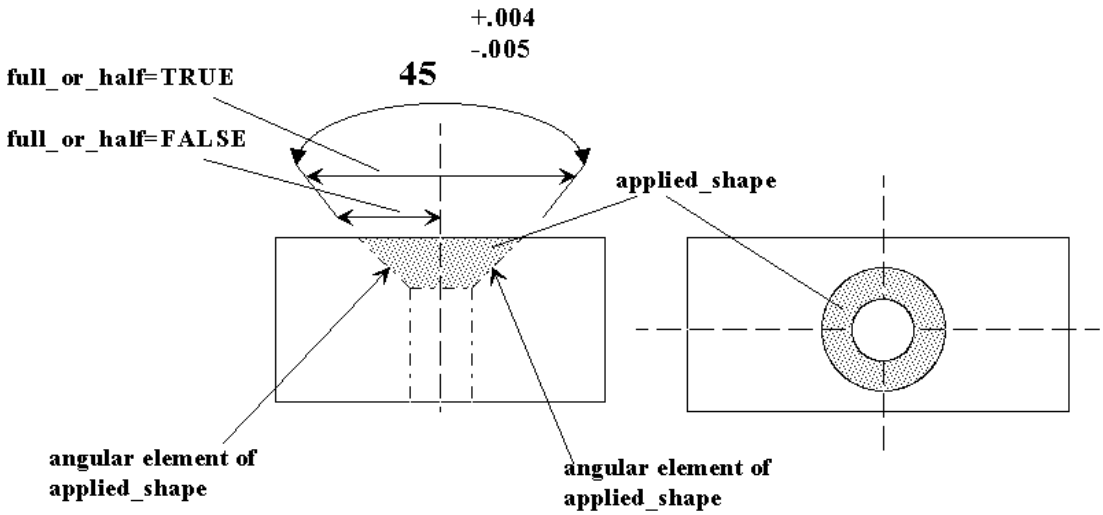


**Figure 3 — Angular_size_dimension_tolerance**

 

The data associated with an Angular_size_dimension_tolerance are the following:

— full_or_half;

— major_angle.

### 4.2.9.1 full_or_half

The full_or_half indicates the method used to establish the Angular_size_dimension_tolerance angle. The angle is either established between the two sides of an angular element or a center line datum and an angular element.

### 4.2.9.2 major_angle

The major_angle specifies the size of the angle for defining the variation. The angle is either the largest or the smallest of the two angles formed by the two elements of the part's shape that are related to the Angular_size_dimension_tolerance.

## 4.2.10 Base_shape

A Base_shape is the initial shape of the material before machining of the features. Each Base_shape is either an Explicit_base_shape_representation (see 4.2.36) or an Implicit_base_shape_representation (see 4.2.60).

NOTE The Base_shape definition is derived from clause 4.2.7 of ISO 10303-224.

## 4.2.11 Block_base_shape

A Block_base_shape is a type of Implicit_base_shape_representation (see 4.2.60) that describes the initial shape of the material as a rectangular cross section of some determined length.

NOTE 1 The Block_base_shape definition is derived from clause 4.2.10 of ISO 10303-224.

NOTE 2 Figure 4 illustrates a Block_base_shape and the associated attributes.

The data associated with a Block_base_shape are the following:

— height;

— width.

### 4.2.11.1 height

The height specifies the size of the side of a Block_base_shape along the Y-axis. See 4.3.13 for the application assertion.

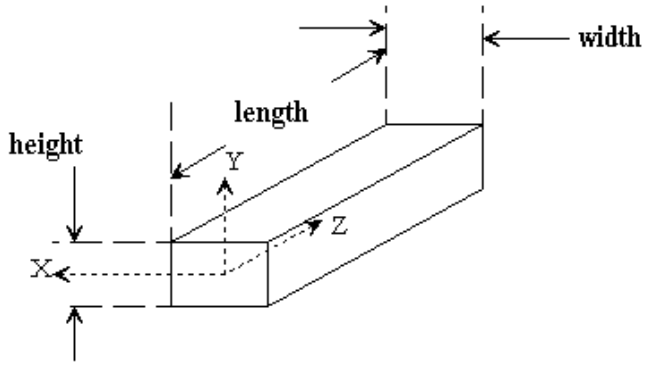


**Figure 4 — Block_base_shape**

## 4.2.11.2 width

The width specifies the size of the side of a Block_base_shape along the X-axis. See 4.3.13 for the application assertion.

## 4.2.12 BSU

A BSU is the identification of 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 4.2.139), a Class_BSU (see 4.2.13), or a Property_BSU (see 4.2.120).

NOTE The BSU definition is derived from clause 4.2.17 of ISO 10303-224.

The data associated with a BSU are the following:

— code.

## 4.2.12.1 code

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

## 4.2.13 Class_BSU

A Class_BSU is a type of BSU (see 4.2.12) 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 shall be unique.

NOTE 3 The Class_BSU definition is derived from clause 4.2.34 of ISO 10303-224.

 

The data associated with a Class_BSU are the following:

— defined_by;

— version.

### 4.2.13.1 defined_by

The defined_by specifies the library supplier who defines the library class. See 4.3.14 for the application assertion.

### 4.2.13.2 version

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

## 4.2.14 Company

A Company is an organization which is responsible for the process plans and products. The data associated with a Company are the following:

— address;

— name.

### 4.2.14.1 address

The address specifies where the Company is located.

### 4.2.14.2 name

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

## 4.2.15 Company_contract_assignment

A Company_contract_assignment is a mechanism to associate a company with a contract, where the assigned company provides information about the contract it is associated to. The data associated with a Company_contract_assignment are the following:

— assigned_company;

— role.

### 4.2.15.1 assigned_company

The assigned_company specifies the company that is used to provide information to the contract. See 4.3.15 for the application assertion.

### 4.2.15.2 role

The assigned_company specifies the purpose for a company fulfilling the role with contracts.

## 4.2.16 Continuous_process

A Continuous_process is the organization of Manufacturing_processes (see 4.2.77) in a sequential order for a specific machine with a specific machine setup of a process plan. The value of the Continuous_process shall be one of the following:

— process_type;

— related_process;

— relating_process.

### 4.2.16.1 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 4.2.16.1.1 to 4.2.16.1.3 for the definition of each allowable value for process_type.

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

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

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

### 4.2.16.2 related_process

The related_process specifies the sequence of process to be processed for a a specific machine with a specific machine setup by the process plan. The sequencing is achieved through chaining together the Manufacturing_process objects (see 4.2.77) The related_process references the next Manufacturing_-process. See 4.3.16 for the application assertion.

### 4.2.16.3 relating_process

The relating_process specifies the sequence of process to be processed for a a specific machine with a specific machine setup by the process plan. The sequencing is achieved through chaining together the Manufacturing_process objects (see 4.2.77) The relating_process references the previous Manufacturing_-process. See 4.3.16 for the application assertion.

 

### 4.2.17 Contract

A Contract is the binding agreement between two parties for the procurement and delivery of goods or services. This agreement is a legally enforceable business arrangement for the supply of certain goods or services by the contractee to the contractor. The data associated with a Contract are the following:

— acquires;

— binding_agreement;

— contract_document;

— contract_number;

— project_number.

#### 4.2.17.1 acquires

The acquires specifies the product that is to be manufactured for this contract. See 4.3.19 for the application assertion.

#### 4.2.17.2 binding_agreement

The binding_agreement specifies the company supplying the contract to manufacture a product. A Company_contract_assignment need not be specified for a particular Contract. See 4.3.17 for the application assertion.

#### 4.2.17.3 contract_document

The contract_document specifies the hardcopy or digital document that details information about the contract. The contract_document need not be specified for a particular Contract. See 4.3.18 for the application assertion.

#### 4.2.17.4 contract_number

The contract_number specifies the unique identification of a Contract within an organization.

#### 4.2.17.5 project_number

The project_number specifies the unique identification, within an organization, that relates the work authorization back to the Contract.

## 4.2.18 Controller

The Controller is the computerized numerical controller that give instructions to a machine for the purpose of creating a product. The data associated with a Controller are the following:

— company_model;

— company_name;

— controller_specification.

### 4.2.18.1 company_model

The company_model specifies the descriptive model identification of the controller.

### 4.2.18.2 company_name

The company_name specifies the descriptive name for the controller.

### 4.2.18.3 controller_specification

The controller_specification specifies the hardcopy or digital document that details information about the controller. See 4.3.20 for the application assertion.

## 4.2.19 Controller_program

An Controller_program is the collection of information, which identifies a specific set of instructions to create a product.

NOTE A program can be stored as a computer file or stored on other media. Program in this context refers to the collection of controller codes necessary to control the machine tools spindle speeds, axis feed rates, and other needed controller functions.

The data associated with an Controller_program are the following:

— accomplished_by;

— controller_data_file;

— controller_type;

— covers;

— id;

— identified_on;

— program_format;

 

— references;

— revision;

— validation_date;

— validation_time.

### 4.2.19.1 accomplished_by

The accomplished_by specifies the process planning activity achieved by the completion of this controller program. The accomplishes need not be specified for a particular Controller_program. See 4.3.23 for the application assertion.

### 4.2.19.2 controller_data_file

The controller_data_file specifies the hardcopy or digital document that is the controller program. The controller_data_file need not be specified for a particular Controller_program. See 4.3.21 for the application assertion.

### 4.2.19.3 controller_type

The controller_type specifies the identification of the manufacturer and model number of the controller attached to the machine tool for which the controller program is written.

### 4.2.19.4 covers

The covers specifies the sequential organization of the process planning activities covered by the controller program. The covers need not be specified for a particular Controller_program. See 4.3.24 for the application assertion.

### 4.2.19.5 id

The id specifies the unique identification, within an organization, of the controller program used for a specific machining sequence specified on the Process_plan_version (see 4.2.116).

### 4.2.19.6 identified_on

The identified_on specifies the product that is having the part shape modified by this controller program. See 4.3.22 for the application assertion.

### 4.2.19.7 program_format

The program_format specifies the identification of the standard used in the data encoding of the controller program.

### 4.2.19.8 references

The references specifies the tooling referenced by the controller program. The references need not be specified for a particular Controller_program See 4.3.25 for the application assertion.

### 4.2.19.9 revision

The revision specifies the unique identification, within an organization, of the change level of the controller program.

### 4.2.19.10 validation_date

The validation_date specifies a calendar date on which the controller program was tested and found to be suitable for production use.

### 4.2.19.11 validation_time

The validation_time specifies the clock time on which the controller program was tested and found to be suitable for production use. The validation_time need not be specified for a particular controller program.

## 4.2.20 Curved_dimension_tolerance

A Curved_dimension_tolerance is a type of Size_tolerance (see 4.2.133) that is the tolerance on a dimension for a curve measured along the entire path of the curve.

NOTE 1The Curved_dimension_tolerance definition is derived from clause 4.2.47 of ISO 10303-224.

NOTE 2 Figure 5 illustrates the Curved_dimension_tolerance.

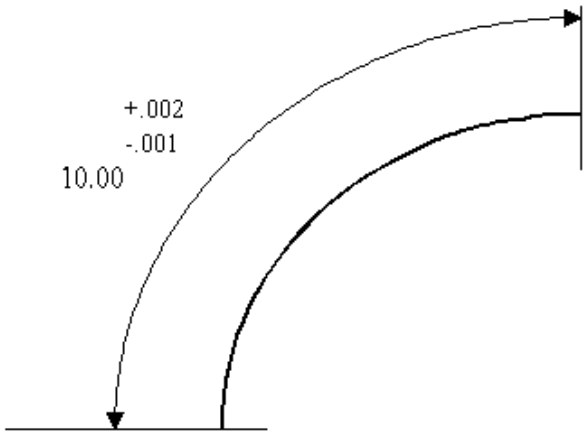


**Figure 5 — Curved_dimension_tolerance**

 

## 4.2.21 Cylindrical_base_shape

A Cylindrical_base_shape is a type of Implicit_base_shape_representation (see 4.2.60) that is the initial shape of the material which is cylindrical.

NOTE 1 The Cylindrical_base_shape definition is derived from clause 4.2.51 of ISO 10303-224.

NOTE 2 Figure 6 illustrates a Cylindrical_base_shape.

The data associated with a Cylindrical_base_shape are the following:

— diameter.

### 4.2.21.1 diameter

The diameter specifies the distance across a Cylindrical_base_shape. See 4.3.26 for the application assertion.

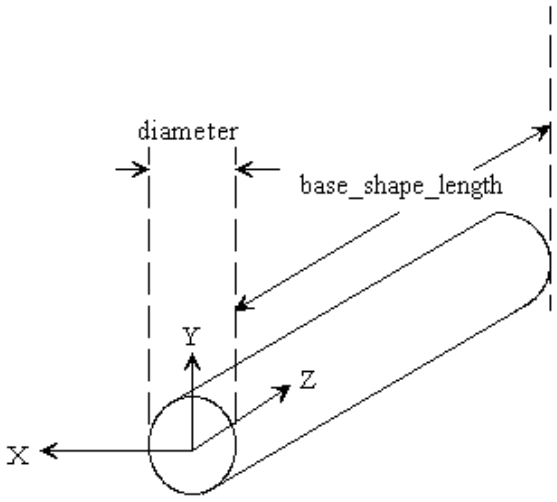


**Figure 6 — Cylindrical_base_shape**

## 4.2.22 Descriptive_parameter

A Descriptive_parameter is a type of Property_parameter (see 4.2.121) that is an explanation of the property being defined by a specification.

NOTE The Descriptive_parameter definition is derived from clause 4.2.61 of ISO 10303-224.

The data associated with a Descriptive_parameter are the following:

— descriptive_string.

### 4.2.22.1 descriptive_string

The descriptive_string specifies a word or group of words by which a Descriptive_parameter is explained.

## 4.2.23 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 The Design_exception_notice definition is derived from clause 4.2.62 of ISO 10303-224.

The data associated with a Design_exception_notice are the following:

— discrepant_part;

— issues;

— issuing_date;

— notice_description;

— notice_number;

— technical_recommendation.

### 4.2.23.1 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 4.3.28 for the application assertion.

### 4.2.23.2 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 4.3.27 for the application assertion.

### 4.2.23.3 issuing_date

The issuing_date specifies the year, month and day when the design_exception_notice was created.

 

### 4.2.23.4 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.

### 4.2.23.5 notice_number

The notice_number specifies a unique identification for each Design_exception_notice.

### 4.2.23.6 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.

## 4.2.24 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 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;

— id;

— identifies;

— revision_level.

### 4.2.24.1 drawing_data_file

The drawing_data_file specifies the hardcopy or digital document that is the drawing. See 4.3.29 for the application assertion.

### 4.2.24.2 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.

### 4.2.24.3 id

The id specifies the unique identification for a Design_reference within an organization.

### 4.2.24.4 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 4.3.30 for the application assertion.

### 4.2.24.5 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.

## 4.2.25 Diameter_dimension_tolerance

A Diameter_dimension_tolerance is a type of Size_tolerance (see 4.2.133) that is the allowable variation of the size of a hole in a surface.

NOTE 1 The Design_exception_notice definition is derived from clause 4.2.64 of ISO 10303-224.

NOTE 2 Figure 7 illustrates the Diameter_dimension_tolerance.

## 4.2.26 Digital_file

A Digital_file is the computer interpretable data. A Digital_file is a type of Document_file (see 4.2.30).

## 4.2.27 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 4.2.67) or a Size_tolerance (see 4.2.133).

NOTE The Dimensional_tolerance definition is derived from clause 4.2.68 of ISO 10303-224.

EXAMPLE A dimension given as 1.624 +.002/-.002 means it may be 1.626 or 1.622, or anywhere between these limit dimensions.

 

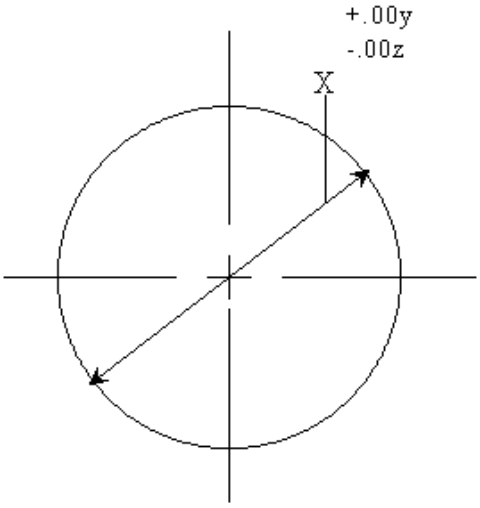


**Figure 7 — Diameter_dimension_tolerance**

The data associated with a Dimensional_tolerance are the following:

— dimension_description;

— dimension_note;

— dimension_value;

— limit;

— significant_digits;

— unit_of_measure.

### 4.2.27.1 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 two places.

### 4.2.27.2 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:

— auxilary;

— theoretical;

— user defined.

NOTE See 4.2.27.2.1 to 4.2.27.2.3 for the definition of each allowable value for limit_qualifier.

**4.2.27.2.1 auxilary:** Restrict auxilary dimension to be a nominal value with no value limitation.

**4.2.27.2.2 theoretical:** Restrict theoretically to be a nominal value with no value limitation.

**4.2.27.2.3 user defined:** a description specified by the user.

### 4.2.27.3 dimension_value

The dimension_value specifies the total amount by which a specific dimension is permitted to vary.

### 4.2.27.4 limit

The limit specifies the tolerance value applied to the Dimension_tolerance. The limit need not be specified for a particular Dimension_tolerance. See 4.3.31 for the application assertion.

### 4.2.27.5 significant_digits

The significant_digits specifies the number of decimal places indicating the accuracy of dimension or tolerance. The Significant_digits need not be specified for a particular Dimensional_tolerance.

### 4.2.27.6 unit_of_measure

The unit_of_measure specifies the unit in which the quantity is expressed.

## 4.2.28 Distance_along_curve_tolerance

A Distance_along_curve_tolerance is a type of Location_tolerance (see 4.2.67) that is the distance calculated between two elements along a path defined by a third element of geometry.

NOTE 1 The Distance_along_curve_tolerance definition is derived from clause 4.2.71 of ISO 10303-224.

NOTE 2 Figure 8 illustrates the Distance_along_curve_tolerance.

 

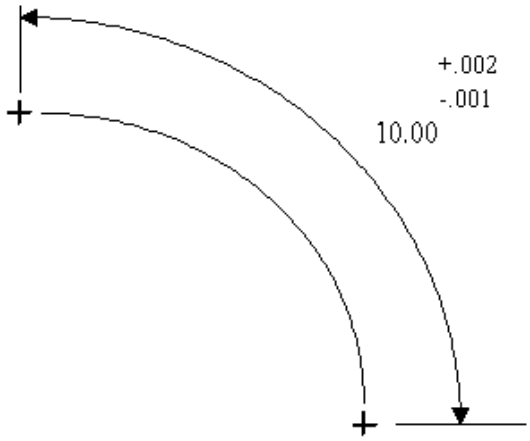


**Figure 8 — Distance_along_curve_tolerance**

The data associated with a Distance_along_curve_tolerance are the following:

— path;

— with_curve_direction.

### 4.2.28.1 path

The path specifies the shape that the tolerance applies to. See 4.3.32 for the application assertion.

### 4.2.28.2 with_curve_direction

The with_curve_direction specifies the direction along the element to apply the tolerance. The tolerance value is either applied from the start point of the curve to the end point of the curve, or the direction does not matter.

## 4.2.29 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 Partial_document_assignment (see 4.2.110). The data associated with a Document_assignment are the following:

— assigned_document;

— role.

### 4.2.29.1 assigned_document

The assigned_document specifies the Document_file (see 4.2.30) that is used to provide information. See 4.3.33 for the application assertion.

### 4.2.29.2 role

The role specifies the meaning of the Document_assignment.

## 4.2.30 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 4.2.56) or a Digital_file (see 4.2.26). The data associated with a Document_file are the following:

— document_properties;

— document_type;

— file_id;

— file_location;

— version_id.

### 4.2.30.1 document_properties

The document_properties specifies the characteristics of the Document_file that relate to the format of the object. See 4.3.34 for the application assertion. The document_properties need not be specified for a particular Document_file (see 4.2.30).

### 4.2.30.2 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;

— geometry;

— NC data;

— process plan;

— sample data.

 

NOTE See 4.2.30.2.1 to 4.2.30.2.6 for the definition of each allowable value for document_type.

**4.2.30.2.1 check plan:** the document represents quality control planning data.

**4.2.30.2.2 drawing:** the document represents a technical drawing.

**4.2.30.2.3 geometry:** the document represents a shape model.

**4.2.30.2.4 NC data:** the document represents numerical control data.

**4.2.30.2.5 process plan:** the document represents process planning data.

**4.2.30.2.6 sample data:** the document represents measured data.

### 4.2.30.3 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.

### 4.2.30.4 file_location

The file_location specifies the information necessary to locate the file in an external storage area. See 4.3.35 for the application assertion. A file_location need not be specified for a particular Document_file.

### 4.2.30.5 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.

## 4.2.31 Document_file_properties

A Document_file_property is the specification of characteristics of a Document_file (4.2.30) that specify the format of the object. At least one of the optional attributes shall be specified for each instance of this object. 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.

### 4.2.31.1 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 UnitedStates of America.

### 4.2.31.2 content_detail_level

The content_detail_level specifies the level of detail that the Document_file (4.2.30) 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 4.2.31.2.1 to 4.2.31.2.2 for the definition of each allowable value for content_detail_level.

**4.2.31.2.1 rough 3d shape:** the 3D shape model without edge rounds and fillets.

**4.2.31.2.2 rounded edges:** the 3D shape model with edge rounds and fillets.

### 4.2.31.3 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 4.2.31.3.1 to for 4.2.31.3.11 the definition of each allowable value for content_geometry_type.

**4.2.31.3.1 2D assembly:** the document contains an assembly structure with reference to the assembled components and their transformation matrices.

**4.2.31.3.2 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 overlayed with the assembly structure for the main components.

**4.2.31.3.3 2D closed volume:** The document contains a 3D shape model in closed body topological surface representation.

**4.2.31.3.4 drawing derived from 3D data:** the document contains a technical drawing that has been derived from a 3D shape model.

**4.2.31.3.5 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.

**4.2.31.3.6 solid model:** the document contains a 3D shape model in advanced boundary representation.

**4.2.31.3.7 solid and surface model:** the document contains a 3D shape model in surface and advanced boundary representation.

**4.2.31.3.8 2D surface model:** the document contains a 3D shape model in surface representation.

**4.2.31.3.9 2D shape:** the document contains a 2D shape model or contours only.

**4.2.31.3.10 2D drawing:** the document contains a technical drawing without 3D shape representation.

**4.2.31.3.11 3D wireframe model:** the document contains a 3D shape model in wireframe representation.

### 4.2.31.4 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.

### 4.2.31.5 creating_interface

The creating_interface specifies the computer application used to create the Document_file (see 4.2.30) 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.

### 4.2.31.6 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.

### 4.2.31.7 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.

### 4.2.31.8 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 4.2.31.8.1 to for 4.2.31.8.10 the definition of each allowable value for data_format.

**4.2.31.8.1 DXF:** the document contains data in Drawing Exchange File format.

**4.2.31.8.2 IGES:** the document contains data in Initial Graphics Exchange Specification format.

**4.2.31.8.3 ISO 10303-203:** the document contains data in ISO 10303-203 format.

**4.2.31.8.4 ISO 10303-214:** the document contains data in ISO 10303-214 format.

**4.2.31.8.5 ISO 10303-224:** the document contains data in ISO 10303-214 format.

**4.2.31.8.6 ISO 10303-238:** the document contains data in ISO 10303-238 format.

**4.2.31.8.7 ISO 6983 :** M and G codes for NC controlers.

**4.2.31.8.8 TIFF CCITT GR4**: the document contains data in TIFF CCITT GR4 format.

**4.2.31.8.9 VDAFS:** the document contains data in VDAFS format.

**4.2.31.8.10 VOXEL:** the document contains data in VOXEL format.

## 4.2.31.9 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 Digital_document (see 4.2.26) or a Document_file (see 4.2.30). 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.

## 4.2.31.10 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 646;

— ISO 3098-1;

— ISO 6937;

— ISO 8859-1;

— ISO 10646.

NOTE See 4.2.31.10.1 to for 4.2.31.10.7 the definition of each allowable value for format_character_code.

**4.2.31.10.1 binary:** the document contains data in binary format.

**4.2.31.10.2 IEC 61286:** the coded character set used to encode the document data according to IEC 61286.

**4.2.31.10.3 ISO 646:** the coded character set used to encode the document data according to ISO 646.

NOTE The character set in ISO 646 is identical to the character set commonly known as ASCII.

**4.2.31.10.4 ISO 3098-1:** the coded character set used to encode the document data is according to ISO 3098-1.

**4.2.31.10.5 ISO 6937:** the coded character set used to encode the document data is according to ISO/IEC6937.

**4.2.31.10.6 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.

**4.2.31.10.7 ISO 10646:** the coded character set used to encode the document data according to ISO/IEC10646.

## 4.2.31.11 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 4.2.56). 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.

 

## 4.2.32 Document_file_relationship

A Document_file_relationship is a relationship between two Document_file (4.2.30) objects. It specifies that the related Document_file is referenced from the relating Document_file. The data associated with a Document_file_relationship are the following:

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'.

— previous_file;

— succeeding_file.

### 4.2.32.1 previous_file

The previous_file specifies the Document_file with the highest precedence. See 4.3.36 for the application assertion.

### 4.2.32.2 succeeding_file

The succeeding_file specifies the Document_file with a lesser precedence. See4.3.36 for the application assertion.

## 4.2.33 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 These Engineering_change_orders apply only to changes that effect process planning.

The data associated with an Engineering_change_order are the following:

— change_order_number;

— new_version.

### 4.2.33.1 change_order_number

The change_order_number specifies a unique identification of the Engineering_change_order.

### 4.2.33.2 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 4.3.37 for the application assertion.

## 4.2.34 Engineering_change_proposal

An Engineering_change_proposal is a document that describes potential modifications to a part.

NOTE The Engineering_change_proposal definition is derived from clause 4.2.74 of ISO 10303-224.

The data associated with an Engineering_change_proposal are the following:

— change_proposal_number;

— incorporated_proposal.

### 4.2.34.1 change_proposal_number

The change_proposal_number specifies a unique identification of the Engineering_change_proposal.

### 4.2.34.2 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 4.3.38 for the application assertion.

## 4.2.35 Executable

The Executable is a type of External_schema_definition (see 4.2.38) that initiate actions on a machine and shall be arranged in a defined order.

NOTE Executable definitions are derived from ISO 14649-10 [1].

The data associated with an Executable the following:

— executable_id.

### 4.2.35.1 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.

 

## 4.2.36 Explicit_base_shape_representation

An Explicit_base_shape_representation is a type of Base_shape (see 4.2.10) that is the geometric representation needed to define the shape of the initial material.

NOTE The Explicit_base_shape_representation definition is derived from clause 4.2.74 of ISO 10303-224.

EXAMPLE A B-rep model containing the geometry for a cast part may be an Explicit_base_shape_representation.

The data associated with an Explicit_base_shape_representation are the following:

— explicit_shape;

— shape_form;

— user_defined_description.

## 4.2.36.1 explicit_shape

The explicit_shape specifies an indicator used to denote the particular shape of the part when it can not be defined with an implicit definition and has an explicit shape.

The value of the explicit_shape shall be one of the following:

— casting;

— composite shape;

— forging;

— user_defined.

NOTE See 4.2.36.1.1 - 4.2.36.1.4 for the definition of each allowable value for explicit_shape.

**4.2.36.1.1 casting**
the base shape of the part is created by pouring molten material into a mold.

**4.2.36.1.2 composite shape**
the base shape of the part is defined with non-metallic materials.

**4.2.36.1.3 forging**
the base shape of the part is created by heating metal and hammering a formed shape.

**4.2.36.1.4 user_defined**
the base shape of the part has non standard description defined by the user.

## 4.2.36.2 shape_form

The shape_form specifies the shape that is the representation of the initial material. See 4.3.39 for the application assertion.

### 4.2.36.3 user_defined_description

The user_defined_description_description specifies a non standard description defined by the user. The user_defined_description_description need not be specified for a Explicit_base_shape_representation.

## 4.2.37 External_file_identification

An External_file_identification is a specification of the location of a file in an external storage system. The data associated with an External_file_id_and_location are the following:

— external_file_id;

— external_file_location.

### 4.2.37.1 external_file_id

The external_id specifies the identifier of a document in an external storage system. A external_id need not be specified for a particular External_file_identification

EXAMPLE An example of an external_file_id may be 'specification.txt'.

### 4.2.37.2 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.

## 4.2.38 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. Each External_schema_definition may be an Executable (see 4.2.35). The data associated with an External_schema_definition are the following:

— reference_schema.

### 4.2.38.1 reference_schema

The reference_schema specifies a file being referenced that conforms to the ISO 10303 schema in digital format. See 4.3.40 for the application assertion.

EXAMPLE An example of a reference_schema is a file that conforms to the ISO 10303-238 schema.

 

## 4.2.39 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. The data associated with an Externally_defined_-representation are the following:

— identified_by;

— location;

— placement.

### 4.2.39.1 identified_by

The identified_by specifies the part being referenced, that is contained in an external library. See 4.3.41 for the application assertion.

### 4.2.39.2 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.

### 4.2.39.3 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.

## 4.2.40 Externally_defined_size_dimension

An Externally_defined_size_dimension is a type of Size_tolerance (see 4.2.133) that is used to identify a size dimension whose definition is provided within an external specification or document. The data associated with an Externally_defined_size_dimension are the following:

— path;

— tolerance_class;

— tolerance_definition.

### 4.2.40.1 path

The path specifies the path along which the Thickness_dimension is applied or measured. See 4.3.42 for the application assertion.

### 4.2.40.2 tolerance_class

The tolerance_class specifies a type of size tolerance that is being defined by in external document.

### 4.2.40.3 tolerance_definition

The tolerance_definition specifies a file being referenced that contains information about the type of tolerance. See 4.3.43 for the application assertion.

## 4.2.41 Feature_dependency

A Feature_dependency is the organization of manufacturing features in a sequential order for process planning. The value of the Feature_dependency shall be one of the following:

— description;

— name;

— related_feature_item;

— relating_feature_item.

### 4.2.41.1 description

The description specifies the word, or group of words, that explain the work performed within the Feature_dependency.

### 4.2.41.2 name

The name specifies the word, or group of words, used to identify the associated value of a specific characteristic of a Feature_dependency.

### 4.2.41.3 related_feature_item

The related_feature_item specifies the sequence of features to be processed by the process plan. The sequencing is achieved through chaining together the Feature_identification_item objects (see 4.2.42). The related_feature_item references the next Feature_identification_item. See 4.3.44 for the application assertion.

### 4.2.41.4 relating_feature_item

The relating_feature_item specifies the sequence of features to be processed by the process plan. The sequencing is achieved through chaining together the Feature_identification_item objects (see 4.2.42). The relating_feature_item references the previous Feature_identification_item. See 4.3.44 and for the application assertion.

 

## 4.2.42 Feature_identification_item

An Feature_identification_item is an identification of machining features associated with a particular activity of the process plan. The data associated with an Feature_identification_item are the following:

— feature;

— feature_relation;

— process.

### 4.2.42.1 feature

The feature specifies the machining feature identified by a Feature_identification_item. See 4.3.46 for the application assertion.

### 4.2.42.2 feature_relation

The feature_relation specifies the interaction between this Feature_identification_item and other Feature_-identification_item objects. A Feature_interaction need not be specified for a particular Feature_-identification_item. See 4.3.45 for the application assertion.

EXAMPLE A feature_relation may be a hole feature having an interaction with a thread feature.

### 4.2.42.3 process

The process specifies the process information being applied the feature. See 4.3.47 for the application assertion.

## 4.2.43 Feature_interaction

A Feature_interaction is the relationship between features that relate to one another for the purpose of process activity planning.

EXAMPLE A Pocket and a Recess in the same pocket may be feature interactions.

The data associated with a Feature_interaction are the following:

— description;

— related_feature.

### 4.2.43.1 description

The description specifies the descriptive information that describes the type of feature interaction.

### 4.2.43.2 related_feature

The related_feature specifies the feature involved in the interaction. The related_feature need not be specified for a particular Feature_interaction. See 4.3.48 for the application assertion.

## 4.2.44 Feature_process

A Feature_process is the assignment of operations and parameters in a Manufacturing_process (see 4.2.77) to a machining feature. The data associated with a Feature_process are the following:

— assigned_operation;

— assigned_process;

— parameter_for_process.

### 4.2.44.1 assigned_operation

The assigned_operations specifies the manufacturing operations assigned to a feature. See 4.3.49 for the application assertion.

### 4.2.44.2 assigned_process

The assigned_process specifies the activities for a a specific machine with a specific machine setup to process the feature. See 4.3.51 for the application assertion.

### 4.2.44.3 parameter_for_process

The parameter_for_process specifies a set of parameters assigned to a feature. See 4.3.50 for the application assertion.

## 4.2.45 Fixture_assembly

A Fixture_assembly is a collection of one or more Fixture_assembly_element objects used to support the part in a known orientation in space. The data associated with a Fixture_assembly are the following:

— assembly_instruction;

— composed_of;

— configuration;

— defined_shape;

 

— documented_by;

— id;

— identified_by.

### 4.2.45.1 assembly_instruction

The assembly_instruction specifies the word, or group of words, that provides directions on how to connect the fixture elements for use as a Fixture_assembly.

### 4.2.45.2 composed_of

The composed_of specifies the Fixture_assembly_element objects (see 4.2.46) or other Fixture_assembly that define a Fixture_assembly. See 4.3.54 and 4.3.53 for the application assertion.

### 4.2.45.3 configuration

The configuration specifies whether the Fixture_assembly is modular or non_modular. A modular assembly is a fixture made up of a group of fixture components, used to support a part. A non_modular assembly is a single component fixture.

### 4.2.45.4 defined_shape

The defined_shape specifies the Shape_aspect (See 4.2.129) that has the shape of the fixture assembly defined by an aspect of geometry. The defined_shape need not be specified for a particular Fixture_ assembly. See 4.3.56 for the application assertion.

### 4.2.45.5 documented_by

The documented_by specifies the fixture referenced by the Design_reference. A Fixture_assembly need not be specified for a particular Fixture_assembly. See 4.3.52 for the application assertion.

### 4.2.45.6 id

The id specifies the unique identification, within an organization, of a fixture used to support a part during the machining process.

### 4.2.45.7 identified_by

The identified_by specifies the Fixture_contract necessary to procure the Fixture_assembly. See 4.3.55 for the application assertion.

## 4.2.46 Fixture_assembly_element

A Fixture_assembly_element is the component or components used in the makeup of a fixture assembly.

NOTE A fixture assembly element can be as simple as a clamp or magnetic table, or as complicated as a modular assembly.

The data associated with a Fixture_assembly_element are the following:

— catalogue_number;

— company_name;

— defined_shape;

— description;

— id;

— name;

— weight.

### 4.2.46.1 catalogue_number

The catalogue_number specifies the unique identification of an item in a fixture catalogue. The catalog_-number need not be specified for a particular Fixture_assembly_element.

### 4.2.46.2 company_name

The company_name specifies the word, or group of words, that uniquely identify the manufacturer of the Fixture_assembly_element.

### 4.2.46.3 defined_shape

The defined_shape specifies the Shape_aspect (See 4.2.129) that has the shape of the Fixture_assembly_element defined by an aspect of geometry. A defined_shape need not be specified for a particular Fixture_assembly_element. See 4.3.57 for the application assertion.

### 4.2.46.4 description

The description specifies the word or group of words used to provide information about the Fixture_-assembly_element and its function.

### 4.2.46.5 id

The id specifies the unique identification, within an organization, of a Fixture_assembly_element.

 

### 4.2.46.6 name

The name specifies the word, or group of words, that identifies the Fixture_assembly_element.

### 4.2.46.7 weight

The weight specifies the numeric value that represents the mass of the Fixture_assembly_element. The weight need not be specified for a particular Fixture_assembly_element.

## 4.2.47 Fixture_contract

A Fixture_contract is the identification of a fixture assembly as the property of the contractee.

NOTE Contract fixtures are established by a fixture_contract associated with a specific fixture or fixture assembly. Fixture_contracts are identified for accounting purposes and billed back to a specific contract. The tool or fixture will have a property tag attached showing it to be the property of the contractee. The company that designed and constructed the tool or fixture charges the contracting company for the tool or fixture and then uses the tool or fixture to produce the parts for the contracting company.

The data associated with a Fixture_contract are the following:

— contract_number;

— design_order;

— fabrication_order;

— fixture_location;

— part_number.

### 4.2.47.1 contract_number

The contract_number specifies the unique identification of a contract associated with a fixture_assembly, within an organization.

### 4.2.47.2 design_order

The design_order specifies the unique identification, within an organization, of the documentation that initiates a request for a design of a contract fixture.

### 4.2.47.3 fabrication_order

The fabrication_order specifies the unique identification, within an organization, of the documentation initiating a request to build the contract fixture.

### 4.2.47.4 fixture_location

The fixture_location specifies the Fixture_assembly (see 4.2.45) that is located by the In_facility_location. See 4.3.58 for the application assertion.

### 4.2.47.5 part_number

The part_number specifies the unique identification, within an organization, of the contract fixture.

## 4.2.48 Fixture_machine_mounting

A Fixture_machine_mounting is a type of Fixture_setup (see 4.2.52) that relates the instructions for the activity of mounting the fixture on the machine. This mounting is by using a predefined alignment of a reference location on the part and a reference location on the machine.

## 4.2.49 Fixture_machine_unmounting

A Fixture_machine_unmounting is a type of Fixture_setup (see 4.2.52) that relates the instructions for the activity of removing the fixture on the machine.

## 4.2.50 Fixture_pallet_mounting

A Fixture_pallet_mounting is a type of Fixture_setup (see 4.2.52) that relates the instructions for the activity of mounting the part on the machine pallet. This mounting is by using a predefined alignment of a reference location on the part and a reference location on the pallet. The data associated with a Fixture_pallet_mounting are the following:

— identifies;

### 4.2.50.1 identifies

The identifies specifies the Machine (see 4.2.68) requiring a Fixture_pallet_mounting. The identifies shall not be specified for a particular Fixture_pallet_mounting. See 4.3.59 for the application assertion.

## 4.2.51 Fixture_pallet_unmounting

A Fixture_pallet_unmounting is a type of Fixture_setup (see 4.2.52) that relates the instructions for the activity of un-mounting the part on the machine pallet. The data associated with a Fixture_pallet_-unmounting are the following:

— identifies;

### 4.2.51.1 identifies

The identifies specifies the Machine (see 4.2.68) requiring a Fixture_pallet_unmounting. The identifies shall not be specified for a particular Fixture_pallet_unmounting. See 4.3.60 for the application assertion.

 

## 4.2.52 Fixture_setup

A Fixture_setup is a type of Setup_activity (see 4.2.128) that is the activity of placing or locating the fixture assembly and the part on the machine prior to performing machining operations. Each Fixture_-setup may be a Fixture_machine_mounting (see 4.2.48), Fixture_machine_unmounting (see 4.2.49), Fixture_pallet_mounting (see 4.2.50) or a Fixture_pallet_unmounting (see 4.2.51).

NOTE Figure 9 illustrates the fixture setup. It illustrates the relationship that exists between the machine, fixture and part. The machine coordinates are shown as Xm, Ym, and Zm. The fixture coordinates are shown as Xf, Yf, and Zf. The part coordinates are shown as Xp, Yp, and Zp.

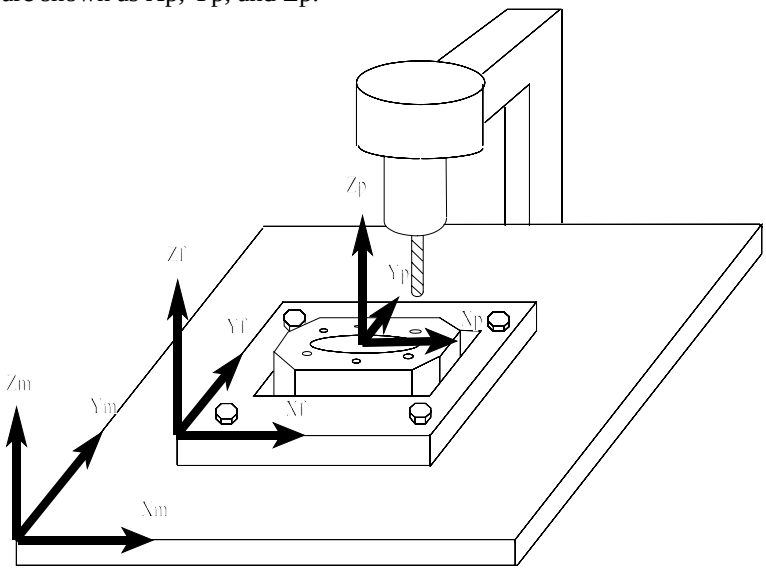


**Figure 9 — Fixture setup**

The data associated with a Fixture_setup are the following:

— identifies;

— identifies_fixture;

— identifies_pallet;

— is_positioned_on.

### 4.2.52.1 is_positioned_on

The is_positioned_on specifies the mounting_position of the fixture. The is_positioned_on need not be specified for a particular Fixture_setup. See 4.3.62 for the application assertion.

### 4.2.52.2 identifies

The identifies specifies the Machine (see 4.2.68) requiring a fixture setup. The identifies need not be specified for a particular Fixture_setup. See 4.3.61 for the application assertion.

### 4.2.52.3 identifies_fixture

The identifies specifies the Fixture_assembly (see 4.2.45) to be setup on a Machine (see 4.2.68). See 4.3.63 for the application assertion.

### 4.2.52.4 identifies_pallet

The identifies_pallet specifies the machine working surface where the part to be manufactured is place. See 4.3.64 for the application assertion.

## 4.2.53 Generic_manufacturing_resource

A Generic_manufacturing_resource is an inventory item used to support machining operations. Each Generic_manufacturing_resource is either a Generic_manufacturing_resource, Resource_with_material (see 4.2.125), or Resource_with_representation (see 4.2.126).

EXAMPLE Durable products such as marking dies, and consumable products such as dye penetrant, coolants, and lubricants are Generic_manufacturing_resources.

The data associated with a Generic_manufacturing_resource are the following:

— description;

— name;

— quantity;

— units.

### 4.2.53.1 description

The description specifies the word, or group of words, that describe the manufacturing 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.

### 4.2.53.2 name

The name specifies the word, or group of words, that make up the unique designation of a manufacturing resource.

### 4.2.53.3 quantity

The quantity specifies the count of the Generic_manufacturing_resource needed.

EXAMPLE Quantity such as ‘three’ in “three gallons”, ‘two’ in “two pair”, or ‘one’ in “one each”.

 

### 4.2.53.4 units

The units specifies the physical measure of the Generic_manufacturing_resource needed.

EXAMPLE Units such as ‘gallons’ in “three gallons”, ‘pair’ as in “two pair”, or ‘each’ as in “one each”.

## 4.2.54 Geometric_model

A Geometric_model is a model containing complete representation of shape.

## 4.2.55 Geometric_tolerance

A Geometric_tolerance is the maximum or minimum variation from true geometric form or position that may be permitted in manufacture.

NOTE Geometric tolerance definitions are derived from ISO 10303-224.

The data associated with a Geometric_tolerance is the following:

— applied_shape.

### 4.2.55.1 applied_shape

The applied_shape specifies the shape on a part that is being toleranced by a Geometric_tolerance. There may be more than one applied_shape for a Geometric_tolerance. See 4.3.65 for the application assertion.

## 4.2.56 Hardcopy

A Hardcopy is the actual stack of paper consisting of one or more sheets, on which some product data is written, printed or plotted.

## 4.2.57 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.

### 4.2.57.1 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.

 

### 4.2.57.2 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.

### 4.2.57.3 nominal

The nominal specifies the nominal value of hardness for a specific material type.

### 4.2.57.4 scale

The scale specifies the method of determining hardness.

EXAMPLE Rockwell and Brinell are examples of scale.

## 4.2.58 Height_dimension

A Height_dimension is a type of Size_tolerance (see 4.2.133) that specifies the size along a straight line that is referred to as height in the referenced shape. The data associated with a Thickness_tolerance are the following:

— path.

### 4.2.58.1 path

The path specifies the path along which the Height_dimension is applied or measured. See 4.3.66 for the application assertion.

## 4.2.59 Illustration

An Illustration is a pictorial presentation used for clarification of an Activity (see 4.2.1) where the narrative alone would not suffice. The data associated with an Illustration are the following:

— clarifies;

— description;

— id;

— is_owned_by.

### 4.2.59.1 clarifies

The clarifies specifies the Activity (see 4.2.1) that requires graphical illustrations as an aid to understanding. A clarifies need not be specified for a particular Illustration. See 4.3.67 for the application assertion.

 

### 4.2.59.2 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.

### 4.2.59.3 id

The id specifies the unique identification, within an organization, of an Illustration.

### 4.2.59.4 is_owned_by

The is_owned_by specifies the View_reference (see 4.2.153) containing the illustration. An View_reference need not be specified for a particular Illustration. See 4.3.68 for the application assertion.

## 4.2.60 Implicit_base_shape_representation

An Implicit_base_shape_representation is a type of Base_shape (see 4.2.10) that is the type of representation needed to define the shape of the initial material. The shape of the material may be either cylindrical, rectangular, or a polygon of any number of sides. Each Implicit_base_shape_representation is either a Block_base_shape (see 4.2.11), Cylindrical_base_shape (see 4.2.21), or a Ngon_base_shape (see 4.2.87).

NOTE 1 The Implicit_base_shape_representation definition is derived from clause 4.2.108 of ISO 10303-224.

NOTE 2 Figure 10 illustrates types of Implicit_base_shape_representations.

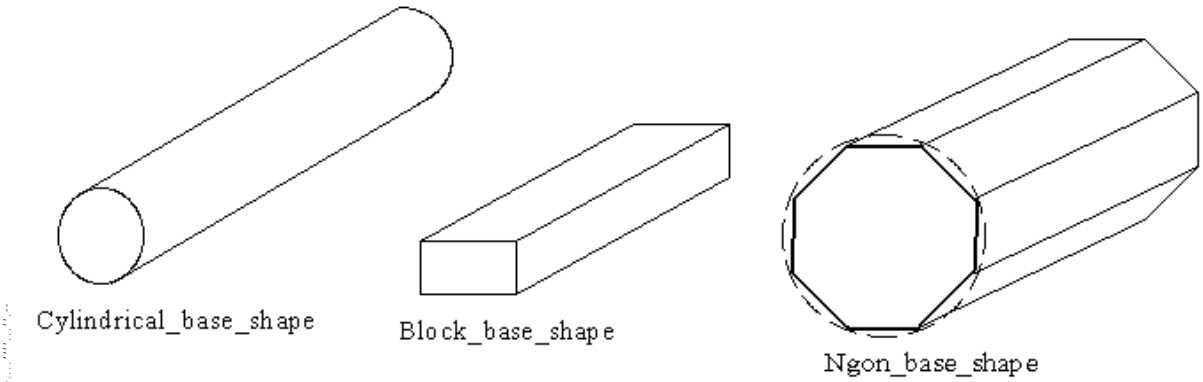


**Figure 10 — Implicit_base_shape_representation**

The data associated with an Implicit_base_shape_representation are the following:

— base_shape_length;

— placement.

### 4.2.60.1 base_shape_length

The base_shape_length is the size of the length of a Implicit_base_shape_representation. See 4.3.69 for the application assertion.

### 4.2.60.2 placement

A placement specifies the positioning of the part with respect to basic material stock. The positioning will be different for different types of Implicit_base_shape_representation. See 4.3.70 for the application assertion.

A Cylindrical_base_shape shall be positioned with the Z axis parallel to the length of the shape. The X and Y axis shall be orthogonal to the Z axis. The axis shall be positioned in the exact center of the circular profile of the Cylindrical_base_shape.

A Block_base_shape shall be positioned with the Z axis parallel to the length of the shape, the Y axis shall be parallel to the height of the shape, and the X axis shall be parallel to the width of the shape. The axis shall be positioned in the exact center of the rectangular profile of the Block_base_shape.

A Ngon_base_shape shall be position with the Z axis parallel to the length of the shape, the X axis shall be parallel to at least one side of the Ngon_base_shape, and the Y axis shall be orthogonal to the X and Z axis. The axis shall be positioned in the exact center of the ngon profile of the Ngon_base_shape.

## 4.2.61 In_facility_location

An In_facility_location is the geographic position, within the enterprise, where a work cell is to be found.

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 sublocation is the hi-bay door.

The data associated with an In_facility_location are the following:

— building_or_area;

— locates;

— location_code;

— sublocation.

### 4.2.61.1 building_or_area

The building_or_area specifies the unique identification, within an organization, of a structure or place.

 

### 4.2.61.2 locates

The locates specifies the Work_Cell (see 4.2.155) that is located by the In_facility_location. See 4.3.71 for the application assertion.

### 4.2.61.3 location_code

The location_code specifies the unique identification, within an organization, of a geographic position within a building.

### 4.2.61.4 sublocation

The sublocation specifies the unique identification, within an organization, of a further refinement of the In_facility_location.

## 4.2.62 Intermediate_shape

A Intermediate_shape is a part shapes that are not the final shape of the part, but an in process shape. This shape allows process planning to define process plan activities that need temporary shape definitions so that the final shape can be producted. The data associated with an Intermediate_shape are the following:

— as_is_shape;

— to_be_shape.

### 4.2.62.1 as_is_shape

The as_is_shape specifies the in process shape of the part. See 4.3.72 for the application assertion.

### 4.2.62.2 to_be_shape

The to_be_shape specifies the final shape of the part. See 4.3.72 for the application assertion.

## 4.2.63 Length_dimension

A Length_dimension is a type of Size_tolerance (see 4.2.133) that specifies the size along a straight line that is referred to as length in the referenced shape.shape.The data associated with a Length_dimension are the following:

— path.

### 4.2.63.1 path

The path specifies the path along which the Length_dimension is applied or measured. See 4.3.73 for the application assertion.

## 4.2.64 Library_part_assignment

An Library_part_assignment is 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.

### 4.2.64.1 definitional_class_BSU

The definitional_class_BSU specifies the identification of the component within a parts library as defined by ISO-13584. See 4.3.74 for the application assertion.

### 4.2.64.2 definitional_property_value_pairs

The definitional_property_value_pairs specifies the set of pairs ( Property_BSU (see 4.2.120), Property_value (See 4.2.122)) defining the properties of the class. A definitional_property_value_pairs need not be specified for a particular Library_part_assignment. See 4.3.75 for the application assertion.

## 4.2.65 Limits_and_fits

A Limits_and_fits is the necessary information to express a tolerance of the limits-and-fits system standardized by ISO 286.

NOTE The Limits_and_fits definition is derived from clause 4.2.112 of ISO 10303-224.

The data associated with a Limits_and_fits are the following:

— deviation;

— fitting type;

— grade.

### 4.2.65.1 deviation

The deviation specifies the class descriptor, by characters, for the designated limets and fits.

NOTE The characters ’A’ to ’ZC’ for holes or ’a’ to ’zc’ for shafts may be used for deviation.

### 4.2.65.2 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 particular Limits_and_fits.

 

### 4.2.65.3 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.

## 4.2.66 Location_dimension_tolerance

A Location_dimension_tolerance is a type of Location_tolerance (see 4.2.67) that is the allowable variation in locating one feature of a part with respect to another.

NOTE 1 The Location_dimension_tolerance definition is derived from clause 4.2.116 of ISO 10303-224.

NOTE 2 Figure 11 illustrates a Location_dimension_tolerance.

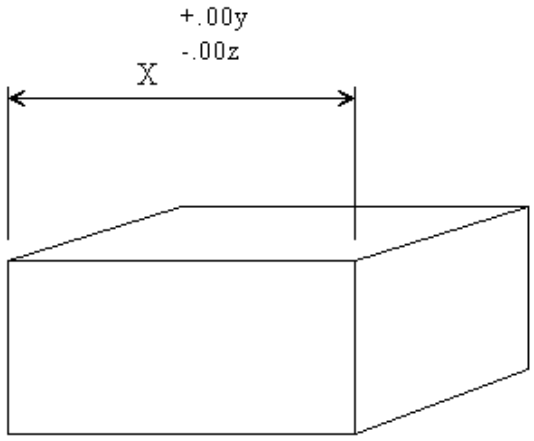


**Figure 11 — Location_dimension_tolerance**

The data associated with a Location_dimension_tolerance are the following:

— plane_and_direction.

### 4.2.66.1 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 4.3.76 for the application assertion.

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

## 4.2.67 Location_tolerance

A Location_tolerance is a type of Dimensional_tolerance (see 4.2.27) 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 4.2.2), Location_dimension_tolerance (see 4.2.66), or Distance_along_curve_tolerance (see 4.2.28).

NOTE The Location_tolerance definition is derived from clause 4.2.118 of ISO 10303-224.

The data associated with a Location_tolerance are the following:

— directed;

— origin_shape;

— termination_shape.

### 4.2.67.1 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.

### 4.2.67.2 origin_shape

The origin_shape specifies the shape on the Part that defines the starting position for a Location_tolerance. See 4.3.77 for the application assertion.

### 4.2.67.3 termination_shape

The termination_shape specifies the shape on the Part that defines the ending position for a Location_-tolerance. See 4.3.77 for the application assertion.

## 4.2.68 Machine

A Machine is a device, with various moving parts, that performs work. The data associated with a Machine are the following:

— company_model;

— company_name;

— contains;

— controlled_by;

 

— id;

— possessed_by.

### 4.2.68.1 company_model

The company_model specifies the unique identification, specified by a machine tool builder, for a set of Machines that all have the same characteristics.

### 4.2.68.2 company_name

The company_name specifies the word, or group of words, that make up the unique designation of the machine tool builder. The name is the machine tool builders name.

### 4.2.68.3 contains

The contains specifies the Pallet associated with this specific Machine. A contains need not be specified for a particular Machine. See 4.3.79 for the application assertion.

### 4.2.68.4 controlled_by

The controlled_by specifies the Controller (see 4.2.18) associated with this specific Machine. A controlled_by need not be specified for a particular Machine. See 4.3.78 for the application assertion.

### 4.2.68.5 id

The id specifies the unique identification of a Machine within an organization.

NOTE This is how the machine is identified on the process plan. It may contain the asset tag number, serial number, or other unique identification.

### 4.2.68.6 possessed_by

The possessed_by specifies the Workstation (see 4.2.156) that contains the Machine. See 4.3.80 for the application assertion.

## 4.2.69 Machine_parameters

A Machine_parameters is the additional attributes for Machine_usage. The data associated with a Machine_parameters are the following:

— axis_range_of_motion;

— axis_range_of_motion_description;

— machining_parameters;

— maximum_feedrate;

— maximum_spindle_speed;

— number_of_control_axis;

— number_of_simultaneous_axis;

— positioning_accuracy;

— spindle_power;

— table_indexing_function;

— table_size.

### 4.2.69.1 axis_range_of_motion

The axis_range_of_motion specifies motion range of each machine tool axis. The axis_range_of_motion need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

EXAMPLE x_axis, y_axis, and z_axis are examples of axis range of motion

### 4.2.69.2 axis_range_of_motion_description

The axis_range_of_motion_description specifies definition for each axis as well as linear or angular measurements. The axis_range_of_motion need not be specified for a particular Machine_parameters. See 4.3.81 for the application assertion.

### 4.2.69.3 machine_parameters

The machine_parameters specifies the machine usage parameters for a specific Machine in a specific Workstation. The machine_parameters need not be specified for a particular Machine_parameters. See 4.3.83 for the application assertion.

### 4.2.69.4 maximum_feedrate

The maximum_feedrate specifies maximum cutting feedrate of machine tool axis. The maximum_feedrate need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

### 4.2.69.5 maximum_spindle_speed

The maximum_spindle_speed specifies maximum speed of machine tool spindle. The maximum_spindle_-speed need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

### 4.2.69.6 number_of_control_axis

The number_of_control_axis specifies number of machine tool axis that can be controlled by numerical controller. The number_of_control_axis need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

 

### 4.2.69.7 number_of_simultaneous_axis

The number_of_simultaneous_axis specifies number of machine tool axis that can be controlled simultaneously by numerical controller. The number_of_simultaneous_axis need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

### 4.2.69.8 positioning_accuracy

The positioning_accuracy specifies positioning accuracy of machine tool axis considering displacement error and repeatability error. The positioning_accuracy need not be specified for a particular Machine_-parameters. See 4.3.82 for the application assertion.

### 4.2.69.9 table_indexing_function

The table_indexing_function specifies table indexing function of machine tool table.If true, machine tool table has table indexing function. If false, it has not. The table_indexing_function need not be specified for a particular Machine_parameters.

### 4.2.69.10 table_size

The table_size specifies size of machine work table. The table_size specifies table diameter in case of circular work table, and specifies table width and table length in case of rectangular work table. The table_size need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

### 4.2.69.11 spindle_power

The spindle_power specifies power of machine tool spindle. The spindle_power need not be specified for a particular Machine_parameters. See 4.3.82 for the application assertion.

## 4.2.70 Machine_setup

A Machine_setup is the activity of preparing machine conditions on the machine prior to performing machining operations. A Machine_setup is a type of Setup_activity (see 4.2.128).

The data associated with a Machine_setup are the following:

— identifies;

— setup_instruction.

### 4.2.70.1 identifies

The identifies specifies the Machine (see 4.2.68) that is being setup for an Activity (see 4.2.1). See 4.3.84 for the application assertion.

### 4.2.70.2 setup_instruction

The setup_instruction specifies the word, or group of words, that explain the work performed to setup the Machine.

## 4.2.71 Machine_usage

A Machine_usage is the identification of a specific machine to be used within a workstation or work_cell. The data associated with a Machine_usage are the following:

— classifies;

— machine_class;

— machine_parameters;

— machine_specification;

### 4.2.71.1 classifies

The classifies specifies the Machine (see 4.2.68) that has a usage within a Workstation. See 4.3.85 for the application assertion.

### 4.2.71.2 machine_class

The machine_class specifies the identification of the category that a specific machine belongs to, such as a horizontal jig mill or a vertical gantry mill.

### 4.2.71.3 machine_specification

The machine_specification specifies additional documentation to define machine usage. A Machine_usage need not be specified for a particular Specification. See 4.3.87 for the application assertion.

### 4.2.71.4 machine_parameters

The machine_parameters specifies the machine usage parameters for a specific Machine in a specific Workstation. See 4.3.86 for the application assertion.

## 4.2.72 Machining_process

A Machining_process is a type of Manufacturing_process (see 4.2.77) that specifies the type of machine, the type of manufacturing setup, and a sequential list of Manufacturing_activity (see 4.2.76) required to perform an automated machining process on a machine. The data associated with a Machining_process are the following:

— required_machine;

 

### 4.2.72.1 required_machine

The required_machine specifies the Machine (see 4.2.68) to be used for a specific machine setup. See 4.3.88 for the application assertion.

## 4.2.73 Machining_tolerance

An Machining_tolerance is the deviation allowed for a cut segment. This is the machining tolerance, not a dimensional tolerance.

NOTE Figure 12 illustrates the three tolerances as they apply to a machine cut sequence.

The data associated with an Machining_tolerance are the following:

— inside_tolerance;

— outside_tolerance;

— total_tolerance.

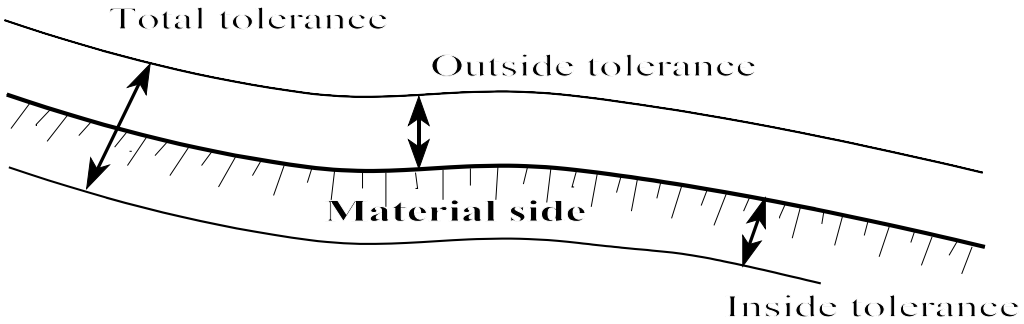


**Figure 12 — Machine cut sequence tolerance**

### 4.2.73.1 inside_tolerance

The inside_tolerance specifies the tolerance band specified and measured from the curve surface in towards the part or material left after the cutter motion along the curve surface.

### 4.2.73.2 outside_tolerance

The outside_tolerance specifies the tolerance band specified and measured from the curve surface away from the part or material left after the cutter motion along the curve surface.

## 4.2.73.3 total_tolerance

The total_tolerance specifies the tolerance band that includes both an inside_tolerance and an outside_-tolerance. A total_tolerance need not be specified for a particular Machining_tolerance.

NOTE The total_tolerance is the inside_tolerance added to the outside_tolerance. The total_tolerance is always a positive value.

## 4.2.74 Manufactured_assembly

A Manufactured_assembly is a type of Part_version (see 4.2.109) that specifies a collection of individual parts or sub-assembly of parts, with orientation. A Manufactured_assembly is considered a sub-assembly when it is a component in another Manufactured_assembly.

NOTE 1 The Manufactured_assembly definition is derived from clause 4.2.121 of ISO 10303-224.

NOTE 2 Figure 13 illustrates the Manufactured_assembly and Mating_definition

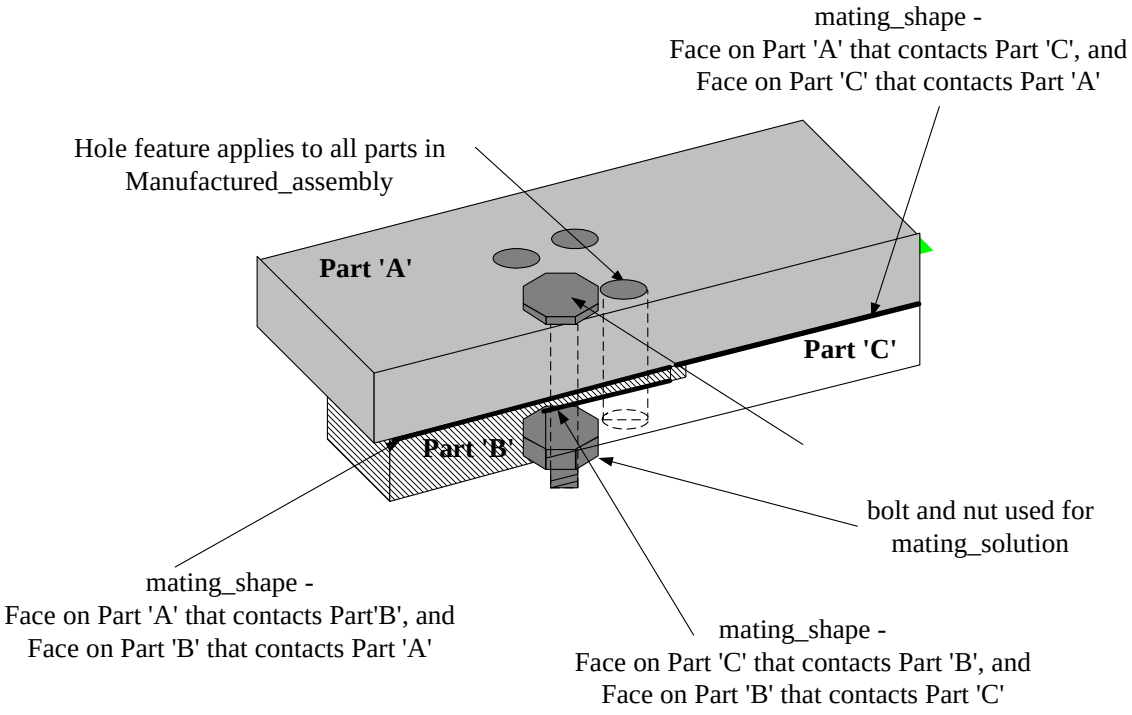


**Figure 13 — Manufactured_assembly and Mating_defintion**

 

## 4.2.75 Manufactured_assembly_relationship

A Manufactured_assembly_relationship is the identification of the Manufactured_assembly, and a component of the assembly. A component is either Single_piece_part (see 4.2.132) or another Manufactured_assembly (see 4.2.74).

NOTE The Manufactured_assembly definition is derived from clause 4.2.122 of ISO 10303-224.

The data associated with a Manufactured_assembly_relationship are the following:

— assembly;

— component;

— orientation.

### 4.2.75.1 assembly

The assembly specifies the Manufactured_assembly (see 4.2.74) that shall have a component of Single_-piece_part (see 4.2.132) or another Manufactured_assembly. See 4.3.89 for the application assertion.

### 4.2.75.2 component

The component specifies either a Single_piece_part (see 4.2.132) or another Manufactured_assembly (see 4.2.74) used to define an assembly. See 4.3.91 for the application assertion.

### 4.2.75.3 orientation

The orientation specifies the transformation of a Part to define its placement in a manufacturing assembly, or sub-assembly. See 4.3.90 for the application assertion.

## 4.2.76 Manufacturing_activity

A Manufacturing_activity is a type of Activity (see 4.2.1) that define any activity that interacts with the part being manufactured. Each Manufacturing_activity is either a Ancillary_activty (see 4.2.6), Part_-mounting (see 4.2.103), Part_routing (see 4.2.106), Part_unmounting (see 4.2.108), Process_activity (see 4.2.114) or Validation (see 4.2.152). The data associated with a Manufacturing_activity are the following:

— micro_plan_reference.

### 4.2.76.1 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 Manufacturing_-activity. See 4.3.92 for the application assertion.

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

## 4.2.77 Manufacturing_process

A Manufacturing_process is either an Machining_process (see 4.2.72), or a Non_machining_process (see 4.2.88) that specifies a sequential list of manufacturing activities to be performed by a specific machine with a specific machine_setup necessary to manufacture a part. The data associated with a Manufacturing_process are the following:

— assigned_feature;

— assigned_operation;

— description;

— in_process_shape;

— name;

— operation_type;

— part_hold_downs;

— setup.

### 4.2.77.1 assigned_feature

The assigned_feature specifies Manufacturing_process_feature (see 4.2.78) that are components of the shape of the Part. The assigned_feature need not be specified for a particular particular Manufacturing_-process. See 4.3.95 for the application assertion.

### 4.2.77.2 assigned_operation

The assigned_operation specifies the Manufacturing_activity (see 4.2.76) that is performed by a specific machine with a specific machine_setup necessary to manufacture a part. See 4.3.94 for the application assertion.

### 4.2.77.3 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.

### 4.2.77.4 in_process_shape

The in_process_shape defines a temporary shape to process that will result in a final part shape. The in_process_shape need not be specified for a particular Manufacturing_process. See 4.3.93 for the application assertion.

### 4.2.77.5 name

The name specifies the word, or group of words, that make up the unique designation of a single setup.

 

### 4.2.77.6 operation_type

The operation_type specifies the specific operation being performed by the Manufacturing_process.

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

— legacy_nc;

— non_nc;

— integrated_nc.

NOTE See 4.2.77.6.1 to 4.2.77.6.3 for the definition of each allowable value for operation_type.

**4.2.77.6.1 legacy_nc:** operation is older type of machining operation that use ISO-6983 standard**.**

**4.2.77.6.2 non_nc:** operation is older type of machining operation that use no NC controller.

**4.2.77.6.3 integrated_nc:** the sequencing of activities and the assignment of features may be performed by the machine controller.

NOTE Integrated_nc shall be using ISO 10303-238.

### 4.2.77.7 part_hold_downs

The part_hold_downs specifies the Part_holding_position (see 4.2.100) that defines the clamping locations and clamp types for holding the part. A part_hold_downs need not be specified for a particular Manufacturing_process. See 4.3.96 for the application assertion.

### 4.2.77.8 setup

The setup specifies the Setup_activity (see 4.2.128) that defines a specific machine setup for a machining process. See 4.3.97 for the application assertion.

## 4.2.78 Manufacturing_process_feature

A Manufacturing_process_feature is the specification of manufacturing features that have been assigned to a planning process. The Manufacturing_process_feature identifies a volume of material that shall be removed to obtain the final part geometry from the initial stock. The shape of the volume are features as defined in ISO 10303-224. The data associated with a Manufacturing_process_feature are the following:

— required_geometry.

### 4.2.78.1 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.

## 4.2.79 Material

A Material is the identification of the raw stock that is the source object the process plan is acting upon.

EXAMPLE Figure 14 illustrates a note block for a part with material.

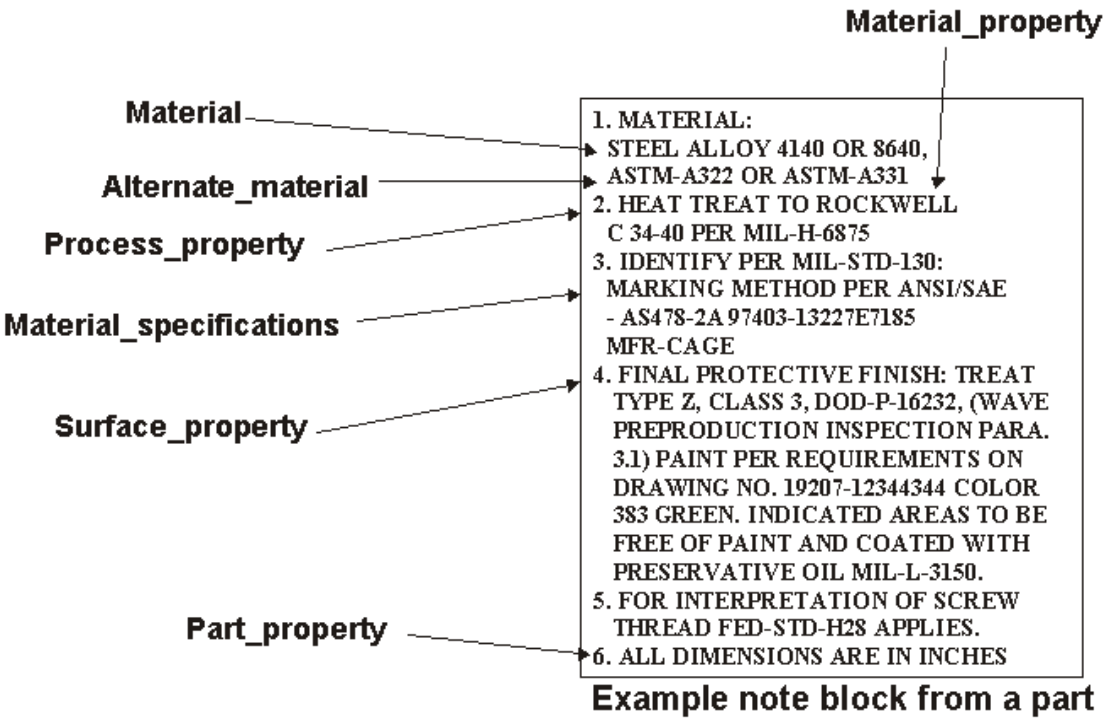


**Figure 14 — Property example**

The data associated with a Material are the following:

— description;

— documented_by;

— identified_by_drawing;

— initial_material_shape;

— material_characteristics;

 

— name;

— order_source;

— source_controlled.

### 4.2.79.1 description

The description specifies the user defined explanation of the material required for the part.

### 4.2.79.2 documented_by

The documented_by specifies the documentation which contains additional information about Material. The material_specification need not be specified for a particular Material. A documented_by need not be specified for a particular Material. See 4.3.99 for the application assertion.

### 4.2.79.3 identified_by_drawing

The identified_by_drawing specifies the Design_reference (see 4.2.24) where the Material information was originally defined. A identified_by_drawing need not be specified for a particular Material. See 4.3.98 for the application assertion.

### 4.2.79.4 initial_material_shape

The initial_material_shape specifies the word, or group of words, that identify the shape of the original stock, or it specifies the implicit or explicit base shape.

EXAMPLE The initial_material_shape includes such things as casting, forging, and bar stock.

See 4.3.100 for the application assertion.

### 4.2.79.5 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 4.3.101 for the application assertion.

### 4.2.79.6 name

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

### 4.2.79.7 order_source

The order_source specifies the word, or group of words, that identify the supplier of the required raw stock.

### 4.2.79.8 source_controlled

The source_controlled specifies the word, or group of words, that identify the preferred manufacturers of the required raw stock.

## 4.2.80 Material_property

A Material_property is the information that describes the properties of the material used to produce a part. The data associated with a Material_property are the following:

— material_hardness;

— material_parameters.

### 4.2.80.1 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 4.3.102 for the application assertion.

### 4.2.80.2 material_parameters

The material_parameters specifies the parameter for the description of Material_property. The material_-parameters need not be specified for a particular Material_property. There may be more than one material_parameters for a Material_property. See 4.3.103 for the application assertion.

## 4.2.81 Material_specification

A Material_specification is the documentation that identifies the mechanical and chemical properties of the initial stock from which the product is produced.

EXAMPLE Figure 14 illustrates a note block for a part with Material_specification.

The data associated with a Material_specification are the following:

— procurement_data.

### 4.2.81.1 procurement_data

The procurement_data specifies the unique identification of the material within an organization. The procurement_data need not be specified for a particular Material_specification.

NOTE This could be a code number, such as “46" or a narrative description.

 

## 4.2.82 Mating_definition

A Mating_definition is a view of a Manufactured_assembly (see 4.2.74) defining the physical connection of two or more Single_piece_part (see 4.2.132) objects. It includes technical information about the kind of connection. This information is independent from the hierarchical assembly structure.

NOTE Figure 13 illustrates the Manufactured_assembly and Mating_definition.

The data associated with a Mating_definition are the following:

— applied_assembly;

— mating_shape;

— mating_solution;

— mating_type.

### 4.2.82.1 applied_assembly

An applied_assembly specifies the Manufactured_assembly (see 4.2.74) that contains the Single_piece_-part (see 4.2.132) objects that have a mating definition. See 4.3.104 for the application assertion.

### 4.2.82.2 mating_shape

The mating_shape specifies the shape that form the area of mating contact between two Single_piece_part (see 4.2.132) objects. See 4.3.105 for the application assertion.

### 4.2.82.3 mating_solution

The mating_solution specifies additional Single_piece_part (see 4.2.132) objects that participate in the Mating_definition. A mating_solution need not be specified for a particular Mating_definition. See 4.3.106 for the application assertion.

EXAMPLE Two parts may be mated together and a nut and bolt used to hold the parts together. The nut and bolt are additional parts used in the mating_solution.

### 4.2.82.4 mating_type

The mating_type specifies the kind of mating, or how the items shall be mated together.

## 4.2.83 Mating_definition_relationship

A Mating_definition_relationship is the additional information about the mating of two particular Single_-piece_part (see 4.2.132) objects that go into a Mating_definition. Two Single_piece_part objects that are referenced by the same Mating_relationship object shall refer to the same Mating_definition. The data associated with a Mating_definition_relationship are the following:

— mated_part;

— mating_part_definition;

— orientation.

### 4.2.83.1 mated_part

The mated_part specifies the Single_piece_part (see 4.2.132) that shall have a mating definition with another single_piece_part. See 4.3.109 for the application assertion.

### 4.2.83.2 mating_part_definition

The mating_part_definition specifies the mating definition for two Single_piece_part (see 4.2.132) objects that contact each other in an assembly. See 4.3.107 for the application assertion.

### 4.2.83.3 orientation

The orientation specifies the transformation of a part to define its placement in a mating definition. A Mating_definition_relationship need not be specified for a particular Mating_definition_relationship. See 4.3.108 for the application assertion.

## 4.2.84 Mating_relationship

A Mating_relationship is the relationship of two Single_piece_part (see 4.2.132) objects that are in the same Manufactured_assembly (see 4.2.74) and are in contact with each other. The data associated with a Mating_relationship are the following:

— predecessor;

— successor.

### 4.2.84.1 predecessor

The predecessor specifies the Single_piece_part (see 4.2.132) with the highest precedence. See 4.3.110 for the application assertion.

### 4.2.84.2 successor

The predecessor specifies the Single_piece_part (see 4.2.132) with the lesser precedence. See 4.3.110 for the application assertion.

 

### 4.2.85 Model_element

A Model_element is a portion of a model representation. The data associated with a Model_element are the following:

— element;

— representation_type.

#### 4.2.85.1 element

The element specifies the portion of the Geometric_model that defines the Model_element. See 4.3.111 for the application assertion.

#### 4.2.85.2 representation_type

The representation_type specifies the geometric representations that define a shape. See 4.3.112 for the application assertion.

### 4.2.86 Mounting_position

The Mounting_position is the placement and orientation for part or fixture mounting. The data associated with a Mounting_position are the following:

— location_origin;

— orientation;

— reference_plane;

#### 4.2.86.1 location_origin

The location_origin specifies a Cartesian point on the machine that establishes the reference for locating the fixture within the machine coordinate system.

#### 4.2.86.2 orientation

The orientation specifies the angle of the fixture that establishes the reference for locating the fixture within the machine coordinate system.

#### 4.2.86.3 reference_plane

The reference_plane specifies a plane on the fixture that establishes the reference for locating the fixture within the machine coordinate system.

## 4.2.87 Ngon_base_shape

An Ngon_base_shape is a type of Implicit_base_shape_representation (see 4.2.60) that specifies the initial shape of the material is a polygon with any number of sides.

NOTE 1 The Ngon_base_shape definition is derived from clause 4.2.136 of ISO 10303-224.

NOTE 2 Figure 15 illustrates a Ngon_base_shape.

The data associated with a Ngon_base_shape are the following:

— circumscribed_or_across_flats;

— corner_radius;

— diameter;

— number_of_sides.

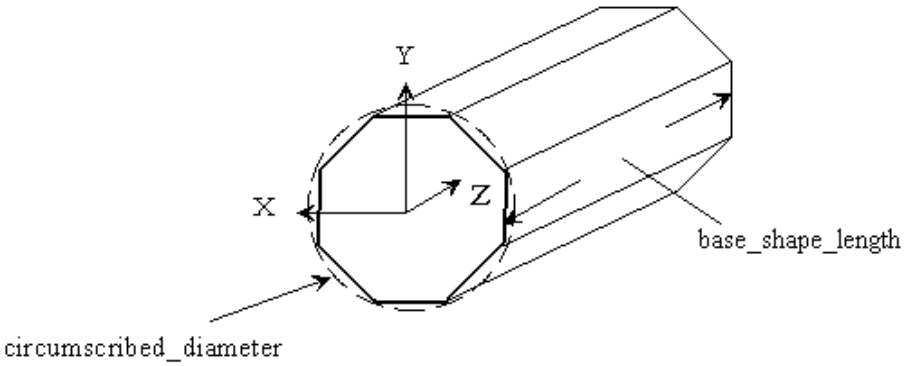


**Figure 15 — Ngon_base_shape**

### 4.2.87.1 circumscribed_or_across_flats

The circumscribed_or_across_flats specifies the type of diameter being used to define the Ngon_profile. Circumscribed is the diameter that the Ngon_base_shape fits inside of, with the corners on the circle that defines the diameter. Across flats, is the diameter that fits inside of the Ngon_profile with the sides of the shape being tangent to the circle that defines the diameter.

### 4.2.87.2 corner_radius

The corner_radius specifies the size of an arc blend between two sides of the ngon. See 4.3.113 for the application assertion.

 

### 4.2.87.3 diameter

The diameter specifies specifies the size of either the circumscribed diameter, or the diameter across the flats. See 4.3.113 for the application assertion.

### 4.2.87.4 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon. See 4.3.113 for the application assertion.

## 4.2.88 Non_machining_process

A Non_machining_process is a type of Manufacturing_process (see 4.2.77) that specifies the type of manufacturing setup, and Manufacturing_activity (see 4.2.76), required to perform a non-automated machining process.

EXAMPLE Examples may be heat treating, painting, or hand grinding.

## 4.2.89 Numeric_parameter

A Numeric_parameter is a type of Property_parameter (see 4.2.121) that is a numeric value with units of the property being defined. A Numeric_parameter is either a Numeric_parameter or a Numeric_-parameter_with_tolerance (see 4.2.90).

NOTE The Numeric_parameter definition is derived from clause 4.2.137 of ISO 10303-224.

The data associated with a Numeric_parameter are the following:

— parameter_unit;

— parameter_value.

### 4.2.89.1 parameter_unit

The parameter_unit specifies the quantity of measure in which the value is given.

EXAMPLE watt, meters, degrees, etc.

### 4.2.89.2 parameter_value

The parameter_value specifies the numeric amount associated with the units of a specific characteristic of interest.

## 4.2.90 Numeric_parameter_with_tolerance

A Numeric_parameter_with_tolerance is a type of Numeric_parameter (see 4.2.89) with an implied tolerance value.

NOTE 1 The Numeric_parameter_with_tolerance definition is derived from clause 4.2.138 of ISO 10303-224.

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.

### 4.2.90.1 implicit_tolerance

The implicit_tolerance specifies the type of tolerance to apply to a numeric parameter value. See 4.3.115, 4.3.116 and 4.3.117 for the application assertion.

## 4.2.91 Object_element_shape_representation

A Object_element_shape_representation is a grouping of representations to define a shape. The data associated with a Object_element_shape_representation are the following:

— representation_type;

— shape_definition.

### 4.2.91.1 representation_type

The representation_type specifies the geometric representations that define a shape. See 4.3.119 for the application assertion.

### 4.2.91.2 shape_definition

The shape_definition specifies the shape that is the representation. See 4.3.118 for the application assertion.

## 4.2.92 Organization

An Organization is the functional group that is responsible for the development, approval, and maintenance of process plans. The data associated with an Organization are the following:

— address_of_site;

— name.

 

### 4.2.92.1 address_of_site

The address_of_site specifies the unique identification of the postal delivery or geographic location of an Organization.

### 4.2.92.2 name

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

## 4.2.93 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. The data associated with an Orientation are the following:

— axis;

— location.

### 4.2.93.1 axis

The axis specifies a line in 3D space about which the part or portions of the part are arranged.

### 4.2.93.2 location

The location specifies a point in 3D space used to position the part or portions of the part.

## 4.2.94 Pallet

A Pallet is a working surface associated with a machine to place a piece part being manufactured. The data associated with an Pallet are the following:

— identifier.

### 4.2.94.1 identifier

The identifier specifies the descriptive name of the Pallet being used by the Machine.

## 4.2.95 Pallet_machine_mounting

A Pallet_machine_mounting is a type of Part_mounting (see 4.2.103) that identifies the instructions for the activity of mounting the pallet to the machine when the part is secured to the pallet. This mounting is accomplished by using a predefined alignment of a reference location on the pallet and a reference location on the machine. The data associated with Part_machine_mounting are the following:

— identifies_pallet;

— identifies_machine.

### 4.2.95.1 identifies_pallet

The identifies_pallet specifies the Pallet (see 4.2.94) that defines the pallet required to hold the part for the process being performed. See 4.3.120 for the application assertion.

### 4.2.95.2 identifies_machine

The identifies_pallet specifies the Machine (see 4.2.68) that defines the machine required to hold the pallet for the process being performed. See 4.3.121 for the application assertion.

## 4.2.96 Pallet_machine_unmounting

A Pallet_machine_unmounting is a type of Part_unmounting (see 4.2.108) that relates the instructions for the activity of removing the pallet on the holding machine.

## 4.2.97 Part_dimensioning_standard

A Part_dimensioning_standard is a type of Document_assignment (see 4.2.29) that is the reference to a document that defines the standard used to define the dimension tolerance used.The data associated with a Part_dimensioning_standard are the following:

— applied_part.

### 4.2.97.1 applied_part

The applied_part specifies the part that uses the dimensioning standard.See 4.3.122 for the application assertion.

## 4.2.98 Part_fixture_mounting

A Part_fixture_mounting is a type of Part_mounting (see 4.2.103) that relates the instructions for the activity of mounting the part on the holding fixture. This mounting is by using a predefined alignment of a reference location on the part and a reference location on the fixture. The data associated with Part_ fixture_mounting is the following:

— identifies_fixture.

NOTE Figure 16 below illustrates the part to fixture relationship that exists during the mounting of the part to a fixture. The fixture coordinates are shown as Xf, Yf, and Zf. The part coordinates are shown as Xp, Yp, and Zp.

 

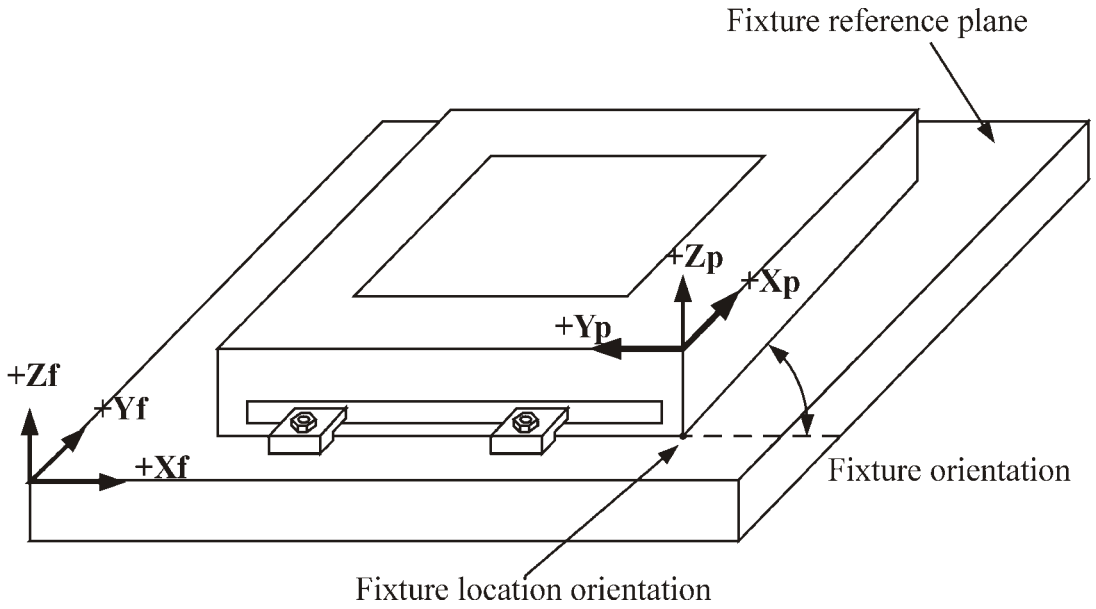


**Figure 16 — Part fixture relationship**

### 4.2.98.1 identifies_fixture

The identifies_fixture specifies the Fixture_assembly (see 4.2.45) that defines the fixture required to hold the part for the process being performed. See 4.3.123 for the application assertion.

## 4.2.99 Part_fixture_unmounting

A Part_fixture_unmounting is a type of Part_unmounting (see 4.2.108) that relates the instructions for the activity of removing the part on the holding fixture.

## 4.2.100 Part_holding_position

A Part_holding_position is the location and type of device being used to hold down the part for manufacturing processes.

The data associated with a Part_holding_position are the following:

— location;

— mating_shape;

— part_hold_downs;

— position_type.

## 4.2.100.1 location

The location specifies where the device is to be position to hold down the part. The location need not be specified for a particular Part_holding_position.

## 4.2.100.2 mating_shape

The mating_shape specifies the shape that form the area of mating contact between two Shape_aspect objects. The mating_shape need not be specified for a particular Part_holding_position. See 4.3.125 for the application assertion.

## 4.2.100.3 part_hold_downs

The part_hold_downs specifies the Fixture_assembly (see 4.2.45) that defines the shape of the device being used to hold the part. See 4.3.124 for the application assertion.

## 4.2.100.4 position_type

The position_type specifies type of device used to hold the part in position.

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

— clamp;

— jack;

— locator.

NOTE See 4.2.100.4.1 to 4.2.100.4.3 for the definition of each allowable value for position_type.

**4.2.100.4.1 clamp:** used for clamping various thickness of material in position, will hold multiple leyers of material.

**4.2.100.4.2 jack:** provides an adjustable rest pad which compensates for work piece irregularities.

**4.2.100.4.3 locator:** removable locating devices for precision alignment of work pieces in jigs and fixtures.

## 4.2.101 Part_machine_mounting

A Part_machine_mounting is a type of Part_mounting (see 4.2.103) that identifies the instructions for the activity of mounting the part to the machine when the part is secured to the machine table. This mounting is accomplished by using a predefined alignment of a reference location on the part and a reference location on the machine.

NOTE Figure 17 illustrates the relationships between the machine and part coordinate systems for mounting a part directly on the machine. The part coordinates are shown as Xp, Yp, and Zp. The machine coordinates are shown as Xm, Ym, and Zm.

 

The data associated with Part_machine_mounting is the following:

— identifies_machine;

### 4.2.101.1 identifies_machine

The identifies_machine specifies the Machine (see 4.2.68) to be used for the setup of a specific Tool_assembly. See 4.3.126 for the application assertion.

## 4.2.102 Part_machine_unmounting

A Part_machine_unmounting is a type of Part_unmounting (see 4.2.108) that relates the instructions for the activity of unmounting the part on the machine.

## 4.2.103 Part_mounting

A Part_mounting is the activity where the material is located in a fixture or on the machine, in preparation for material removal. A Part_mounting is a type of Manufacturing_activity (see 4.2.76). Part_mounting may be either a Pallet_machine_mounting (see 4.2.95), Part_fixture_mounting (see 4.2.98), or Part_-machine_mounting (see 4.2.101). The data associated with Part_mounting is the following:

— is_located_on;

— is_positioned_on;

— part_location_origin.

### 4.2.103.1 is_located_on

The is_located_on specifies the Part_shape (see 4.2.107) which defines the shape of the part the fixture is holding. The is_located_on need not be specified for a particular Part_mounting. See 4.3.128 for the application assertion.

### 4.2.103.2 is_positioned_on

The is_positioned_on specifies the mounting_position of the fixture. See 4.3.127 for the application assertion.

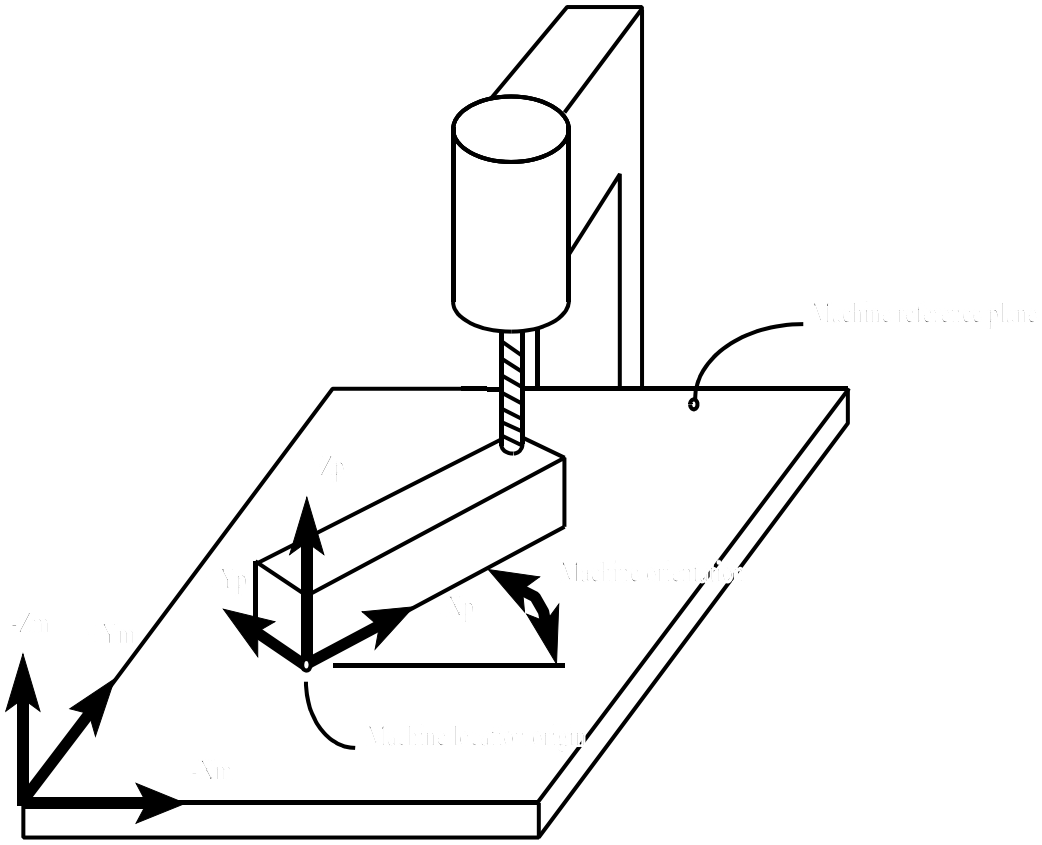

**Figure 17 — Part machine relationship**

## 4.2.103.3 part_location_origin

The part_location_origin specifies a cartesian point on the part that establishes the reference for locating the part within another coordinate system.

NOTE The part_location_origin is used to align the part in the machine coordinate system for part_machine_mounting loading and in the fixture coordinate system for part_fixture_mounting loading.

## 4.2.104 Part_placement

The Part_placement is the transformation of part shape from the originating orientation to define manufacturing assembly and sub-assembly. The data associated with a Part_placement are the following:

— originating_orientation;

— oriented_physical_form;

— resulting_orientation.

 

### 4.2.104.1 originating_orientation

The originating_orientation specifies the orientation of a part or a sub-assembly prior to being positioned in an assembly or another sub_assembly. See 4.3.129 for the application assertion.

### 4.2.104.2 oriented_physical_form

The oriented_physical_form specifies the shape of a part or sub-assembly that is being re-positioned. See 4.3.130 for the application assertion.

### 4.2.104.3 resulting_orientation

The resulting_orientation specifies the orientation of a part or sub-assembly in an assembly or another sub_assembly. See 4.3.129 for the application assertion.

## 4.2.105 Part_property

A Part_property is a specific characteristic about the form, fit, or function of a part.

EXAMPLE Figure 14 illustrates a note block for a part with Part_property.

NOTE The Part_property definition is derived from clause 4.2.150 of ISO 10303-224.

The data associated with a Part_property are the following:

— part_characteristics.

### 4.2.105.1 part_characteristics

The part_characteristics specifies the parameter to describe the Part_property. The part_characteristics need not be specified for a particular Part_property. There may be more than one part_characteristic for a Part_property. See 4.3.131 for the application assertion.

## 4.2.106 Part_routing

A Part_routing is a type of Manufacturing_activity (see 4.2.76) that is the activity in which the material or product is relocated from one discrete location to another. The data associated with Part_routing is the following:

— method.

### 4.2.106.1 method

The method specifies the description of the technique used to move the part.

## 4.2.107 Part_shape

A Part_shape is the physical form of the part that is being machined. The data associated with a Part_-shape are the following:

— base_shape_definition;

— elements;

— representation_form.

### 4.2.107.1 base_shape_definition

The base_shape_definition specifies either the implicit or the explicit definition, or the initial shape of the material before machining of the Part. A base_shape_definition need not be specified for a particular Part_shape. See 4.3.132 for the application assertion.

### 4.2.107.2 elements

The elements specifies the components of the shape of the Part. The element need not be specified for a particular Shape. See 4.3.134 for the application assertion.

### 4.2.107.3 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 4.3.133 for the application assertion.

## 4.2.108 Part_unmounting

A Part_unmounting is a type of Manufacturing_activity (see 4.2.76) that is the activity where the material is removed from a fixture or machine after an operation has been completed. Each Part_unmounting may be a Part_fixture_unmounting (see 4.2.99), Pallet_machine_unmounting (see 4.2.96), Part_machine_-unmounting (see 4.2.102).

## 4.2.109 Part_version

A Part_version is the unique identification, within an organization, of a formal change in the design of a part. Each Part_version may be one of the following: Manufactured_assembly (see 4.2.74), or Single_-piece_part (see 4.2.132). The data associated with a Part_version are the following:

— alternate_plan;

— documented_by;

— effectivity;

— fabrication_defined_by;

 

— nomenclature;

— part_number;

— physical_form;

— property_characteristics;

— revision_level.

### 4.2.109.1 alternate_plan

The alternate_plan specifies the Alternate_process_plan (see 4.2.5)that defines an alternative option for the process plan. A alternate_plan need not be specified for a particular Part_version. See 4.3.135 for the application assertion.

### 4.2.109.2 documented_by

The documented_by specifies the product referenced by the Design_reference. A documents_version need not be specified for a particular Part_version. See 4.3.136 for the application assertion.

### 4.2.109.3 effectivity

The effectivity specifies the identification of a specific process plan that is applied to an instance of the manufacture of a part.

### 4.2.109.4 fabrication_defined_by

The fabrication_defined_by specifies the Part_version (See 4.2.109) created as a result of the process plan. See 4.3.138 for the application assertion.

### 4.2.109.5 nomenclature

The nomenclature specifies the word, or group of words by which a particular part is commonly known within an organization.

### 4.2.109.6 part_number

The part_number specifies the unique identification of a product within an organization.

NOTE This identification is not to be confused with the identification of an instance or occurrence of an item. This identification is the association of the item with the design.

### 4.2.109.7 physical_form

The physical_form specifies the Part_shape (see 4.2.107) that defines the shape of the part to be manufactured. See 4.3.137 for the application assertion.

### 4.2.109.8 property_characteristics

The property_characteristics specifies the Property (see 4.2.119) that defines the associated data about the physical structure of the part. A property_characteristics need not be specified for a particular Part_version. See 4.3.139 for the application assertion.

### 4.2.109.9 revision_level

The revision_level specifies the unique identification, within an organization, of the version of a part.

## 4.2.110 Partial_document_assignment

A Partial_document_assignment is a type Document_assignment (see 4.2.29) that defines a restriction on the application of information defined within document. The data associated with a Partial_document_-assignment are the following:

— document_portion.

### 4.2.110.1 document_portion

The document_portion specifies the word or group of words that convey the subject or sub contents of the Document.

## 4.2.111 Performance_rate

A Performance_rate is the time it takes to perform an Activity. Each Performance_rate may be either an Allowed_time (see 4.2.2) or a Production_rate (see 4.2.118). The data associated with a Performance_rate is the following:

— allowed_type;

— source.

### 4.2.111.1 allowed_type

The allowed_type specifies the word or group of words that identify the item within the category of allowed times. An Performance_rate 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.

### 4.2.111.2 source

The source specifies the origin of the Performance_rate, such as standards, policies and procedures, or estimates.

## 4.2.112 Planning_group_member

A Planning_group_member is a person, within an organization, responsible for performing a process

 

planning activity.

NOTE Functions performed include such activities as process planning, checking, approving and releasing the process plans.

The data associated with Planning_group_member are the following:

— department;

— employed_by;

— id;

— name;

— phone_number.

### 4.2.112.1 department

The department specifies the unique identification, within an organization, of a group that performs the process planning activity.

### 4.2.112.2 employed_by

The employed_by specifies the Organization (See 4.2.92) that employes the planning group member. See 4.3.140 for the application assertion.

### 4.2.112.3 id

The id specifies the unique identification of a Planning_group_member within an organization.

### 4.2.112.4 name

The name specifies the word, or group of words, that make up the unique designation of a Planning_-group_member and can include first name and surname.

NOTE Enterprise-specific policies or procedures will dictate the format and usage of the name, such as Joe Smith, J Smith, or just Smith.

### 4.2.112.5 phone_number

The phone_number specifies the unique identification of a telephone device within an area. The phone_-number need not be specified for a particular Planning_group_member.

NOTE A telephone device can be any equipment that communicates over the telephone lines such as a modem, FAX machine, or telephone. Phone_numbers can include country code, area code, and telephone number.

## 4.2.113 Plus_minus_value

The Plus_minus_value 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 The Plus_minus_value definition is derived from clause 4.2.168 of ISO 10303-224.

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.

### 4.2.113.1 lower_limit

The lower_limit specifies the low limit value.

### 4.2.113.2 significant_digits

The significant_digits specifies the number of decimal places indicating the accuracy of a dimension or tolerance.

### 4.2.113.3 upper_limit

The upper_limit specifies the high limit value.

## 4.2.114 Process_activity

The Process_activity is a type of Manufacturing_activity (see 4.2.76) ) that define the processing activity to interacts with the material part being manufactured.

EXAMPLE Examples of processing activities are material removal, heat treating, painting, or oiling.

The data associated with a Process_activity are the following:

— described_by;

— identifies;

— identifies_fixture;

— process_parameters;

 

— tolerances;

— type_of_operation;

— uses_to_perform.

### 4.2.114.1 described_by

The described_by specifies the Shape_aspect (see 4.2.129) that defines a portion of the part shape being affected by the Process_activity. A described_by need not be specified for a particular Shape_aspect. See 4.3.145 for the application assertion.

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

### 4.2.114.2 identifies

The identifies specifies the Machine_setup (see 4.2.70)that defines the positoning of the part on the machine for the process being performed. A identifies need not be specified for a particular Machine_setup. See 4.3.142 for the application assertion.

### 4.2.114.3 identifies_fixture

The identifies_fixture specifies the Fixture_assembly (see 4.2.45) that defines the fixture required to hold the part for the process being performed. A identifies_fixture need not be specified for a particular Fixture_assembly. See 4.3.141 for the application assertion.

### 4.2.114.4 process_parameters

The process_parameters specifies the Process_property (see 4.2.117) that define additional process information. The process_parameters need not be specified for a particular Process_activity. See 4.3.144 for the application assertion.

### 4.2.114.5 type_of_operation

The type_of_operation specifies the type of process being performed.

EXAMPLE Examples of processes may be Finishing_process, Material_removal, Tube_bending, Fabrication_-process, Conditioning, Welding_process, Assembly, Forming.

### 4.2.114.6 tolerances

The tolerances specifies the machining tolerance required for the manufacturing process. A tolerances need not be specified for a particular Process_activity. See 4.3.143 for the application assertion.

### 4.2.114.7 uses_to_perform

The uses_to_perform specifies the Tool_assembly (see 4.2.145) that defines the tool required to remove magerial from the part. A uses_to_perform need not be specified for a particular Process_activity. See 4.3.146 for the application assertion.

## 4.2.115 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.

### 4.2.115.1 classification_date

The classification_date specifies the calendar date the security classification was granted.

### 4.2.115.2 declassification_date

The declassification_date specifies the calendar date the security classification was revoked or canceled.

### 4.2.115.3 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 4.3.147 for the application assertion.

### 4.2.115.4 identified_by_process_plan

The identified_by_process_plan specifies the Process_plan_version (see 4.2.116) that requires security information. The identified_by_process_plan shall be specified for a particular Process_plan_security. See 4.3.148 for the application assertion.

### 4.2.115.5 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.

 

## 4.2.116 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.

NOTE A discussion of Process_plan_version is given in Annex L.

The data associated with a Process_plan_version are the following:

— activities_to_produce_part;

— auxiliary_header_information;

— description;

— feature_dependency_suggestion;

— id;

— manufacturing_GT_code;

— process_plan_information;

— quantity_of_parts;

— quantity_range;

— required_material.

### 4.2.116.1 activities_to_produce_part

The activities_to_produce_part specifies the Manufacturing_process (see 4.2.77) that define the sequential list of activities controlled by the process plan. See 4.3.150 for the application assertion.

### 4.2.116.2 auxiliary_header_information

The auxiliary_header_information specifies the Property_parameter (see 4.2.121) that define the additional information about the process plan that is to be exchanged or archived. In many cases this information is necessary for Computer Aided Process Planning (CAPP) systems to interface with a Database Management System (DBMS). See 4.3.152 for the application assertion.

### 4.2.116.3 description

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

### 4.2.116.4 feature_dependency_suggestion

The feature_dependency_suggestion specifies the Feature_dependency (see 4.2.41) that defines the feature hierarchical structure for volume removal. A feature_dependency_suggestion need not be specified for a particular Process_plan_version. See 4.3.149 for the application assertion.

### 4.2.116.5 id

The id specifies the unique identification, within an organization, of an Process_plan_version.

### 4.2.116.6 manufacturing_GT_code

The manufacturing_GT_code specifies the unique identification, within an organization, of the product shape and product manufacturing characteristics based upon an established classification schema. The manufacturing_GT_code need not be specified for a particular Process_plan_version.

### 4.2.116.7 process_plan_information

The process_plan_information specifies the Special_instruction (see 4.2.135) additional information for the process plan. A process_plan_information need not be specified for a particular Process_plan_version. See 4.3.154 for the application assertion.

### 4.2.116.8 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.

### 4.2.116.9 quantity_range

The quantity_range specifies the Range_of_parts (see 4.2.124)that define a range of parts that may be manufactured per the process plan. The quantity_range need not be specified for a particular Process_plan_version. See 4.3.153 for the application assertion.

EXAMPLE An example of a quantity_range may be 10 to 100 parts for this process plan.

### 4.2.116.10 required_material

The required_material specifies the Material (see 4.2.79) that is to be used to manufacture the product for this process plan. See 4.3.151 for the application assertion.

 

## 4.2.117 Process_property

A Process_property is the characteristics of a series of actions or operations directed toward changing the part.

NOTE The Process_property definition is derived from clause 4.2.172 of ISO 10303-224.

EXAMPLE Figure 14 illustrates a note block for a part with Process_property.

The data associated with a Process_property are the following:

— process_name;

— process_characteristics.

### 4.2.117.1 process_name

The process_name specifies a word or group of words by which a Process_property is commonly referred.

### 4.2.117.2 process_characteristics

The process_characteristics specifies the parameter to describe the Process_property. The process_-characteristics need not be specified for a particular Process_property. There may be more than one process_characteristic for a Process_property. See 4.3.155 for the application assertion.

## 4.2.118 Production_rate

A Production_rate is the time it takes to produce a measurable amount of items. A Production_rate is a type of Performance_rate (see 4.2.111).

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.

### 4.2.118.1 time_per_unit

The time_per_unit specifies the duration for completing one task.

### 4.2.118.2 unit_quantity

The unit_quantity specifies the number of units to be produced.

## 4.2.119 Property

A Property is a characteristic associated with the physical structure or integrity of an element of a part.

NOTE The Property definition is derived from clause 4.2.178 of ISO 10303-224.

The data associated with a Property are the following:

— material_characteristics;

— part_property_characteristics;

— process_characteristics;

— property_characteristics;

— property_description;

— property_name;

— surface_characteristics.

### 4.2.119.1 material_characteristics

The material_characteristics specifies the information that describe material for manufacturing the Part. The material_characteristics need not be specified for a particular Property. There may be more than one material_characteristics for a Property. See 4.3.156 for the application assertion.

### 4.2.119.2 part_property_characteristics

The part_property_characteristics specifies the information that describe properties of the Part. The part_-property_characteristics need not be specified for a particular Property. There may be more than one part_property_characteristics for a Property. See 4.3.157 for the application assertion.

### 4.2.119.3 process_characteristics

The process_characteristics specifies information that describe processes for manufacturing the part. The process_characteristics need not be specified for a particular Property. There may be more than one process_characteristics for a Property. See 4.3.158 for the application assertion.

### 4.2.119.4 property_characteristics

The property_characteristics specifies information that describe properties of the Part. The property_-characteristics need not be specified for a particular Property. There may be more than one property_-characteristics for a Property. See 4.3.159 for the application assertion.

 

### 4.2.119.5 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 4.3.160 for the application assertion.

### 4.2.119.6 property_name

The property_name specifies a word or group of words by which a property is commonly referred.

### 4.2.119.7 surface_characteristics

The surface_characteristics specifies information that describe surface conditions of the Part. The surface_characteristics need not be specified for a particular Property. There may be more than one surface_characteristics for a Property. See 4.3.161 for the application assertion.

## 4.2.120 Property_BSU

A Property_BSU is a type of BSU (see 4.2.12) that identifies a property basic semantical unit of a class in a parts library.

NOTE The Property_BSU definition is derived from clause 4.2.179 of ISO 10303-224.

The data associated with a Property_BSU are the following:

— name_scope;

— version.

### 4.2.120.1 name_scope

The name_scope specifies the class this property belongs to. See 4.3.162 for the application assertion.

### 4.2.120.2 version

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

## 4.2.121 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 4.2.22) or a Numeric_parameter (see 4.2.89).

NOTE The Property_parameter definition is derived from clause 4.2.180 of ISO 10303-224.

The data associated with a Property_parameter are the following:

— parameter_name.

### 4.2.121.1 parameter_name

The parameter_name specifies a word or group of words that identify a characteristic of interest for a Property_parameter.

## 4.2.122 Property_value

A Property_value is a value for a property as specified in the property basic semantical unit. The value type is specified in subtypes.

NOTE The Property_value definition is derived from clause 4.2.181 of ISO 10303-224.

The data associated with a Property_value are the following:

— property_BSU;

— value_amount.

### 4.2.122.1 property_BSU

The property_BSU specifies the Property_BSU (see 4.2.120) that defines the basic semantical unit. See 4.3.163 for the application assertion.

### 4.2.122.2 value_amount

The value_amount specifies the value that is defined as a boolean, integer, number, logical, string, or real.

## 4.2.123 Radial_dimension_tolerance

A Radial_dimension_tolerance is a type of Size_tolerance (see 4.2.133) that is the allowable variation for the radial distance from the center of a circular curve to a point on the curve.

NOTE 1 The Radial_dimension_tolerance definition is derived from clause 4.2.183 of ISO 10303-224.

NOTE 2 Figure 18 illustrates the Radial_dimension_tolerance.


**Figure 18 — Radial_dimension_tolerance**

 

## 4.2.124 Range_of_parts

A Range_of_parts is the upper and lower allowable range of parts that can be manufactured for a specified process plan. The data associated with a Range_of_parts are the following:

— high_value;

— low_value.

### 4.2.124.1 high_value

The high_value specifies the highest allowable value for a process plan.

### 4.2.124.2 low_value

The low_value specifies the lowest allowable value for a process plan.

## 4.2.125 Resource_with_material

A Resource_with_material is a type of Generic_manufacturing_resource (see 4.2.53) that provides the means to reference material information about a resource. The data associated with a Resource_with_-material are the following:

— resource_material.

### 4.2.125.1 resource_material

The resource_material specifies the Material (see 4.2.79)that is required as a resource material to manufacture a part. See 4.3.164 for the application assertion.

## 4.2.126 Resource_with_representation

A Resource_with_representation is a type of Generic_manufacturing_resource (see 4.2.53) that provides the means to reference information about a resource within a parts library dictionary. The data associated with a Resource_with_representation are the following:

— resource_documentaton.

### 4.2.126.1 resource_documentaton

The resource_documentaton specifies the Externally_defined_representation (see 4.2.39). See 4.3.165 for the application assertion.

## 4.2.127 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.

### 4.2.127.1 approved_by

The approved_by specifies the Status_authority (see 4.2.137) that defines approval of the revision. See 4.3.167 for the application assertion.

### 4.2.127.2 description

The description specifies the word, or group of words, that explain the Revision.

### 4.2.127.3 reason_for_revision

The reason_for_revision specifies the word, or group of words, that explain the need for the change.

### 4.2.127.4 related_to

The related_to specifies the Process_plan_version (see 4.2.116) that is the successor process plan. See 4.3.166 for the application assertion.

### 4.2.127.5 relating_to

The relating_to specifies the Process_plan_version (see 4.2.116) that is the predecessor process plan. See 4.3.166 for the application assertion.

### 4.2.127.6 revision_level

The revision_level specifies the unique identification, within a process plan or activity, of the change level.

 

## 4.2.128 Setup_activity

A Setup_activity is a type of Activity (see 4.2.1) that defines preparing the fixture, machine, tool, or others for a manufacturing process. Each Activity is either an Fixture_setup (see 4.2.52), Machine_setup (see 4.2.70), Tool_setup (see 4.2.151), or a Ancillary_setup (see 4.2.7).

## 4.2.129 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. The data associated with a Shape_aspect are the following:

— element;

— representation_form;

— representation_shape.

### 4.2.129.1 element

The element specifies Manufacturing_process_feature (see 4.2.78) that are components of the shape of the Part. The element need not be specified for a particular Shape_aspect. See 4.3.168 for the application assertion.

### 4.2.129.2 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 4.3.170 for the application assertion.

### 4.2.129.3 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 4.3.169 for the application assertion.

## 4.2.130 Shape_aspect_representation

A Shape_aspect_representation is a grouping of geometric elements with respect to the representation of a part. The data associated with a Shape_aspect_representation are the following:

— shape_definition.

### 4.2.130.1 shape_definition

The shape_definition specifies the Geometric_model (see 4.2.54) to define the shape that is the representation. See 4.3.171 for the application assertion.

## 4.2.131 Shape_representation_type

A Shape_representation_type is the types of geometric representations that may be used to define product shape. The data associated with a Shape_representation_type are the following:

— geometry_type.

## 4.2.131.1 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;

— faceted_b_rep;

— manifold_surface_with_topology;

— non_topological_surface_and_wireframe;

— wireframe_with_topology.

NOTE See 4.2.131.1.1 to 4.2.131.1.5 for the definition of each allowable value for feature_type.

**4.2.131.1.1 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.

**4.2.131.1.2 faceted_b_rep:** the geometric representation of the shape, or an aspect of the shape of a part, by a faceted boundary representation solid model. The geometric representation allows for the definition of shapes represented by planar surfaces as the bounding surfaces. Only points and planar polygons are used in this representation. Much of the topology information is implicit for this representation. Shells consist of faces bounded exclusively by polygons.

**4.2.131.1.3 manifold_surface_with_topology:** the geometric representation of the shape, or an aspect of the shape of a part, using manifold surfaces with topology. 3D curves, surfaces, and topology are used to define the outer boundary of the part.

**4.2.131.1.4 non_topological_surface_and_wireframe:** the geometric representation of the shape, or an aspect of the shape of a part, using surface or wireframe geometry without topology. These representations are formed by the use of points, curves and surfaces only. The boundaries of the curves are defined explicitly by points on the curves and explicit associations between the points and the curves that they bound. The boundaries of the surfaces are defined by curves on the surfaces and explicit associations between the curves and the surfaces that they bound. Surfaces and curves must be explicitly trimmed unless they are closed.

 

**4.2.131.1.5 wireframe_with_topology:** the geometric representation of the shape, or an aspect of the shape of a part, using wireframes that define an implicit volume or are trimmed by edge topology. This includes 3D curves and topology that define a graph of vertices and edges.

## 4.2.132 Single_piece_part

A Single_piece_part is a type of Part_version (See 4.2.109) that is the physical item which is intended to be produced through the manufacturing process.

NOTE The Single_piece_part definition is derived from clause 4.2.214 of ISO 10303-224.

The data associated with a Single_piece_part are the following:

— alternate_material_definition;

— material_definition;

## 4.2.132.1 alternate_material_definition

The alternate_material_definition specifies the secondary material choices of raw stock for producing the Part. There may be more than one alternate_material_definition for a part. An alternate_material_definition need not be specified for a particular Single_piece_part. See 4.3.172 for the application assertion.

## 4.2.132.2 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 4.3.173 for the application assertion.

## 4.2.133 Size_tolerance

A Size_tolerance is a type of Dimensional_tolerance (see 4.2.27) that is the size dimension tolerance characteristic for a geometric element. Each Size_tolerance is either an Angular_size_dimension_tolerance (see 4.2.9), Curved_dimension_tolerance (see 4.2.20), Diameter_dimension_tolerance (see 4.2.25), Radial_dimension_tolerance (see 4.2.123), Length_dimension (see 4.2.63), Width_dimension (see 4.2.154), Height_dimension (see 4.2.58), Externally_defined_size_dimension (see 4.2.40), or a Thickness_tolerance (see 4.2.141).

NOTE The Size_tolerance definition is derived from clause 4.2.215 of ISO 10303-224.

The data associated with a Size_tolerance are the following:

— applied_shape;

— envelope.

## 4.2.133.1 applied_shape

The applied_shape specifies the physical shape of the Part that is being toleranced. See 4.3.174 for the application assertion.

### 4.2.133.2 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 indicate that the envelope is or is not required for the Size_-tolerance.

## 4.2.134 Special_capability

A Special_capability is the identification of a machine's ability to perform unique or specialized activities during the production process.

EXAMPLE Special_capability includes such things as the ability to handle large material sizes and weights.

The data associated with a Special_capability are the following:

— name;

— possessed_by;

— specification.

### 4.2.134.1 name

The name specifies the word, or group of words, that make up the unique designation of the Special_-capability.

### 4.2.134.2 possessed_by

The possessed_by specifies the Workstation (See 4.2.156) that requires additional information about capabilities. See 4.3.175 for the application assertion.

### 4.2.134.3 specification

The specification specifies the documentation that provide the details of the specific capability and its operating parameters. The specification need not be specified for a particular Special_capability.

## 4.2.135 Special_instruction

A Special_instruction is a description of some aspect of either an Activity (see 4.2.1) or an Process_plan_version (see 4.2.116) that requires further explanation be provided to the individual performing the corresponding Activity.

NOTE 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.

### 4.2.135.1 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 must be met or observed.

### 4.2.135.2 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.

## 4.2.136 Specification

A Specification is a document that defines information pertaining to properties or processes for a part or an aspect of a part. The data associated with a Specification are the following:

— description;

— number_id;

— revision;

— specification_document;

— subclass;

— title.

### 4.2.136.1 description

The description specifies the word, or group of words, that relate detail information about the Specification. A description need not be specified for a particular Specification.

### 4.2.136.2 number_id

The number_id specifies the unique identification of a Specification within an organization.

### 4.2.136.3 revision

The revision specifies the unique identification of the Specification release level.

### 4.2.136.4 specification_document

The specification_document specifies the Document_assignment (see 4.2.29) that is used to provide information. A specification_document need not be specified for a particular Specification. See 4.3.176 for the application assertion.

### 4.2.136.5 subclass

The subclause specifies the identification of a subordinate section within a Specification.

### 4.2.136.6 title

The title specifies the word or phrase by which the Specification is known or referenced.

## 4.2.137 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.

### 4.2.137.1 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.

### 4.2.137.2 date

The date specifies the calendar date on which the approval_title was given.

### 4.2.137.3 is_given_by

The is_given_by specifies the Planning_group_member (see 4.2.112) who has the authority to approve process plans. See 4.3.177 for the application assertion.

 

## 4.2.138 Supplemental_document

A Supplemental_document is a type of Specification (see 4.2.136) 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 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. These procedures are considered to be external to the process planning function, and are directly related to business practice in the enterprise.

## 4.2.139 Supplier_BSU

A Supplier_BSU (supplier basic semantical unit) is a type of BSU (see 4.2.12) that identifies the supplier of a parts library.

NOTE The Supplier_BSU definition is derived from clause 4.2.228 of ISO 10303-224.

## 4.2.140 Surface_property

A Surface_property is the characteristics of a surface that are elements of the shape of a part.

EXAMPLE Figure 14 illustrates a note block for a part with Surface_property.

NOTE The Surface_property definition is derived from clause 4.2.229 of ISO 10303-224.

The data associated with a Surface_property are the following:

— surface_characteristics;

— surface_finish.

## 4.2.140.1 property_characteristics

The surface_characteristics specifies the parameter to describe the Surface_property. The surface_-characteristics need not be specified for a particular Surface_property. See 4.3.178 for the application assertion.

## 4.2.140.2 surface_finish

The surface_finish specifies indicates a type of Surface_property is a surface finish. The Surface_property is either a surface finish property,or is for other surface properties.

## 4.2.141 Thickness_tolerance

A Thickness_tolerance is a type of Size_tolerance (see 4.2.133) that represents a thickness. A Thickness may be represented with only an applied_shape or it may also have a specified path to be measured along.

EXAMPLE 1 The Thickness_tolerance may be the remaining thickness below a blind hole, the hole bottom is specified as an applied shape, the remaining part shape is specified as the used path.

EXAMPLE 2 The Thickness_tolerance may be the thickness of a coating layer which is the applied shape, with no path of measurement.

The data associated with a Thickness_tolerance are the following:

— path.

### 4.2.141.1 path

The path specifies the path along which the Thickness_dimension is applied or measured. See 4.3.179 for the application assertion.

## 4.2.142 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 4.2.27.3) shall be a tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.89.2) shall be a tolerance value. There shall be a qualifier that describes the tolerance context.

NOTE The Tolerance_limit definition is derived from clause 4.2.241 of ISO 10303-224.

The data associated with a Tolerance_limit are the following:

— limit_qualifier.

### 4.2.142.1 limit_qualifier

The limit_qualifier specifies a description of the Tolerance_limit context. The values of the limit_qualifier may be one of the following:

— maximum;

— minimum;

— user defined.

NOTE See 4.2.142.1.1 to 4.2.142.1.3 for the definition of each allowable value for limit_qualifier.

 

**4.2.142.1.1 maximum:** the upper limit on a dimension.

**4.2.142.1.2 minimum:** the lower limit on a dimension.

**4.2.142.1.3 user defined:** a limit type specified by the user.

## 4.2.143 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 4.2.27.3) may be a nominal tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.89.2) may be a nominal tolerance value.

NOTE The Tolerance_range definition is derived from clause 4.2.242 of ISO 10303-224.

The data associated with a Tolerance_range are the following:

— lower_range;

— significant_digits;

— upper_range.

### 4.2.143.1 lower_range

The lower_range specifies the lowest allowable value for a dimensional tolerance.

### 4.2.143.2 significant_digits

The significant_digits specifies the number of decimal places indicating the accuracy of the tolerance.

### 4.2.143.3 upper_range

The upper_range specifies the highest allowable value for a dimensional tolerance.

## 4.2.144 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 The Tolerance_value definition is derived from clause 4.2.243 of ISO 10303-224.

The data associated with a Tolerance_value are the following:

— defined_value.

### 4.2.144.1 defined_value

The defined_value specifies the tolerance deviation value. See 4.3.180, 4.3.181, 4.3.182, and 4.3.183 for the application assertion.

## 4.2.145 Tool_assembly

A Tool_assembly is the collection of Tool_assembly_elements (see 4.2.146). A Tool_assembly can be a single Tool_assembly_element.

EXAMPLE Tool_assembly includes such items as the cutter and its components. It can also include such items as tool inserts, the tool collet and tool holder.

The data associated with Tool_assembly are the following:

— composed_of;

— defined_shape;

— documented_by;

— id;

— identified_in_contract;

— identifying_data;

— number_of_spares;

— tool_parameters.

### 4.2.145.1 composed_of

The composed_of specifies the subassembly of Tool_assembly_element (see 4.2.146) components that define the assembly of the Tool_assembly. See 4.3.187 for the application assertion.

### 4.2.145.2 defined_shape

The defined_shape specifies the Shape_aspect (See 4.2.129) that has the shape of the tool assembly defined by an aspect of geometry. A defined_shape need not be specified for a particular Tool_assembly. See 4.3.186 for the application assertion.

### 4.2.145.3 documented_by

The documented_by specifies the tooling referenced by the Design_reference. The documented_by need not be specified for a particular Tool_assembly. A documents need not be specified for a particular Tool_assembly. See 4.3.184 for the application assertion.

 

### 4.2.145.4 id

The id specifies the unique identification of a Tool_assembly within an organization.

### 4.2.145.5 identified_in_contract

The identified_in_contract specifies the Tool_contract (see 4.2.148) that defines procurement of the tool. A identified_in_contract need not be specified for a particular Tool_assembly. See 4.3.188 for the application assertion.

### 4.2.145.6 identifying_data

The identifying_data specifies the additional tool data for tool placement information. A tool_placement_instruction need not be specified for a particular Tool_assembly. See 4.3.189 for the application assertion.

### 4.2.145.7 number_of_spares

The number_of_spares specifies the number of spare tools_assembly objects required for the process plan activity. A number_of_spares need not be specified for a particular Tool_assembly.

### 4.2.145.8 tool_parameters

The tool_parameters specifies the Property_parameter (see 4.2.121) that is a property being defined for a specific Tool_assembly. A tool_parameters need not be specified for a particular Tool_assembly. See 4.3.185 for the application assertion.

## 4.2.146 Tool_assembly_element

A Tool_assembly_element is a component of the tool assembly.

NOTE For simple tools the Tool_assembly_element can be the tool assembly itself.

The data associated with a Tool_assembly_element are the following:

— catalogue_number;

— company_name;

— defined_shape;

— description;

— id;

— name;

— tool_property;

— tool_representation.

### 4.2.146.1 catalogue_number

The catalogue_number specifies the identification of the catalogue from which the Tool_assembly_element was ordered. The catalogue_number need not be specified for a particular Tool_assembly_element.

### 4.2.146.2 company_name

The company_name specifies the word or phrase that constitutes the distinctive designation of the Tool_-assembly_element manufacturer.

### 4.2.146.3 defined_shape

The defined_shape specifies the Shape_aspect (See 4.2.129) that has the shape of the tool assembly defined by an aspect of geometry. A defined_shape need not be specified for a particular Tool_assembly_element. See 4.3.186 for the application assertion.

### 4.2.146.4 description

The description specifies the word, or group of words, that describe the Tool_assembly_element. A description need not be specified for a particular Tool_assembly_element.

### 4.2.146.5 id

The id specifies the unique identification of the Tool_assembly_element within an organization.

### 4.2.146.6 name

The name specifies the word, or group of words, that describes the Tool_assembly_element.

EXAMPLE The name can include such words as end mill, slab cutter, and counterbore.

### 4.2.146.7 tool_property

The tool_property specifies the Specificaton (see 4.2.136) to define the external document that contains any additional information that may be required to create the Tool_assembly. The tool_property need not be required for a Tool_assembly_element. See 4.3.192 for the application assertion.

### 4.2.146.8 tool_representation

The tool_representation specifies the Externally_defined_representation (see 4.2.39) to define the external part library information pertaining to a Tool_assembly. A tool_representation need not be specified for a particular Tool_assembly_element. See 4.3.190 for the application assertion.

 

## 4.2.147 Tool_body

A Tool_body is a Tool_assembly_element (see 4.2.146) that represents the main portion of the tool assembly. The data associated with a Tool_body are the following:

— tool_body_parameter.

## 4.2.147.1 tool_body_parameter

The tool_body_parameter specifies additional information about the Tool_body. A tool_body_parameter need not be specified for a particular Tool_body. See 4.3.193 for the application assertion.

## 4.2.148 Tool_contract

A Tool_contract is the identification of a tool assembly as the property of the contractee.

NOTE The contract tools are identified as tool assemblies with an associated Tool_contract. They are identified as such for accounting proposes and billed back to a specific contract.

The data associated with a Tool_contract are the following:

— contract_document;

— contract_number;

— design_order;

— fabrication_order;

— part_number.

## 4.2.148.1 contract_document

The contract_document specifies the Specificaton (see 4.2.136) to define the external document that contains any additional information that may be required by the contract. See 4.3.194 for the application assertion.

## 4.2.148.2 contract_number

The contract_number specifies the unique identification, within an organization, of a tool contract.

## 4.2.148.3 design_order

The design_order specifies the unique identification, within an organization, of the documentation initiating a request for the design of a contract tool.

### 4.2.148.4 fabrication_order

The fabrication_order specifies the unique identification, within an organization, of the documentation that initiates a request to build the contract tool.

### 4.2.148.5 part_number

The part_number specifies the unique identification, within an organization, of a contract tool.

## 4.2.149 Tool_placement_instruction

A Tool_placement_instruction is the tool information required by the programmer. This includes any special instructions that may need to be communicated to the operator via controller console display commands.

EXAMPLE The Tool_placement_instruction includes such information as cutter geometry, turret, and tool magazine.

The data associated with a Tool_placement_instruction are the following:

— configures;

— instruction_document;

— tool_position.

### 4.2.149.1 configures

The configures specifies the Tool_magazine_turret_carousel (see 4.2.150) that requires additional tool information. See 4.3.196 for the application assertion.

### 4.2.149.2 instruction_document

The instruction_document specifies the Specificaton (see 4.2.136) to defines the external document that contains any special instructions that must be conveyed to the machine operator. See 4.3.195 for the application assertion.

### 4.2.149.3 tool_position

The tool_position specifies the unique location within an machine tool’s turret, magazine, or carousel.

 

### 4.2.150 Tool_magazine_turret_carousel

A Tool_magazine_turret_carousel is the holder of the tool assemblies for machining operations. The tool magazine or carousel is preloaded in accordance with the instructions for sequencing the cutters. The data associated with a Tool_magazine_turret_carousel are the following:

— id;

— tool_capacity;

— used_in

### 4.2.150.1 id

The id specifies the unique identification of a tool magazine, turret, or carousel within an organization.

### 4.2.150.2 tool_capacity

The tool_capacity specifies the maximum number of tool assemblies that the magazine, turret, or carousel is capable of holding.

### 4.2.150.3 used_in

The used_in specifies the Workstation (See 4.2.156) that contains the tool holder. See 4.3.197 for the application assertion.

### 4.2.151 Tool_setup

A Tool_setup is a type of Setup_activity (see 4.2.128) that is the activity of placing or locating the tool assembly on the machine prior to performing machining operations. The data associated with a Tool_setup are the following:

— identifies_machine;

— identifies_tool;

— tool_placement.

### 4.2.151.1 identifies_machine

The identifies_machine specifies the Machine (see 4.2.68) to be used for the setup of a specific Tool_assembly. See 4.3.126 for the application assertion.

### 4.2.151.2 identifies_tool

The identifies_tool specifies the Tool_assembly (see4.2.145) to be used for a Manufacturing_process. See 4.3.199 for the application assertion.

### 4.2.151.3 tool_placement

The tool_placement specifies the Tool_magazine_turret_carousel(see 4.2.150) to define where to place the Tool_assembly to be used for a Manufacturing_process. See 4.3.200 for the application assertion.

## 4.2.152 Validation

A Validation is used to identify both on-machine and off-machine production verification requirements. This is typically referred to as in-process inspection, and may be used to describe any automated or manual measurement requirements. A Validation is a type of Manufacturing_activity (see 4.2.76). The data associated with a Validation are the following:

— checks;

— identifies;

— method_of_validation;

— uses_for_validation;

— uses_to_perform.

### 4.2.152.1 checks

The checks specifies the Controller_program (see4.2.19) used in the manufacture of the product. A Validation need not be specified for a particular Controller_program to be specified. See 4.3.201 for the application assertion.

### 4.2.152.2 identifies

The identifies specifies the Fixture_assembly (see 4.2.45) used in the manufacture of the product. A Validation need not be specified for a particular Fixture_assembly to be specified. See 4.3.202 for the application assertion.

### 4.2.152.3 method_of_validation

The method_of_validation specifies the method used in validation the part.

EXAMPLE Examples of validation methods may be in process inspection or dimensional measuring machine.

### 4.2.152.4 uses_for_validation

The uses_for_validation specifies the Part_shape (see 4.2.107) which represents the as designed shape to validate against. See 4.3.203 for the application assertion.

### 4.2.152.5 uses_to_perform

The uses_to_perform specifies the Tool_assembly (see4.2.145) used in the manufacture of the product. See 4.3.204 for the application assertion.

 

## 4.2.153 View_reference

A View_reference is the unique identification, within a drawing, of a specific area of interest or view referenced by an activity. The data associated with a View_reference are the following:

— identifies;

— owned_by;

— sheet;

— view;

— zone.

### 4.2.153.1 identifies

The identifies specifies the Activity (see 4.2.1) referencing the View_reference for the purpose of planning a machining activity. A identifies need not be specified for a particular View_reference. See 4.3.205 for the application assertion.

### 4.2.153.2 owned_by

The owned_by specifies the Design_reference (see 4.2.24) containing the view. An Design_reference need not be specified for a particular View_reference. See 4.3.206 for the application assertion.

### 4.2.153.3 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 need not be specified for a particular View_reference.

### 4.2.153.4 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.

### 4.2.153.5 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.

## 4.2.154 Width_dimension

A Width_dimension is a type of Size_tolerance (see 4.2.133) that specifies the size along a straight line that is referred to as width in the referenced shape.shape.The data associated with a Width_dimension are the following:

— path.

### 4.2.154.1 path

The path specifies the path along which the Width_dimension is applied or measured. See 4.3.207 for the application assertion.

## 4.2.155 Work_cell

A Work_cell is the identification of an area within the enterprise, consisting of one or more workstations. A Work_cell can include the necessary auxiliary equipment required to produce a completed part. The data associated with a Work_cell are the following:

— contains;

— description;

— id;

— referenced_by.

### 4.2.155.1 contains

The contains specifies the Workstation (See 4.2.156) contained in the Work_cell. See 4.3.209 for the application assertion.

### 4.2.155.2 description

The description specifies the word, or group of words, that explain the work performed within the Work_cell.

### 4.2.155.3 id

The id specifies the unique identification of a Work_cell.

### 4.2.155.4 referenced_by

The referenced_by specifies the Activity (see 4.2.1) referencing the Work_cell for the purpose of planning a machining activity. See 4.3.208 for the application assertion.

 

### 4.2.156 Workstation

A Workstation is the identification of an area where work is performed. It can include the storage space required for the material being worked on. The data associated with a Workstation are the following:

— description;

— id.

### 4.2.156.1 description

The description specifies the word, or group of words, that explain the work performed within the Workstation. The description need not be specified for a particular Workstation.

### 4.2.156.2 id

The id specifies the unique alphanumeric text string assigned to a Workstation.

## 4.3 Application assertions

### 4.3.1 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.

### 4.3.2 Activity to Generic_manufacturing_resource

Each Activity has the resources defined by zero, one, or many Generic_manufacturing_resource objects. Each Generic_manufacturing_resource used_in the resources for zero, one, or many Activity objects.

### 4.3.3 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.

### 4.3.4 Activity to Material

Each Activity has the references defined by zero or one Material objects. Each Material defines the is_resource_for for exactly one Activity objects.

### 4.3.5 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.

### 4.3.6 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.

### 4.3.7 Activity to Supplemental_document

Each Activity has the constrained_by defined by zero, one, or many Supplemental_document objects. Each Supplemental_document defines the constrained_by for zero, one or many Activity objects.

### 4.3.8 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.

### 4.3.9 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.

### 4.3.10 Alternate_material to Material

Each Alternate_material has the material_substitute defined by exactly one Material. Each Material is the material_substitute for zero, one, or many Alternate_material objects.

### 4.3.11 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.

### 4.3.12 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.

### 4.3.13 Block_base_shape to Numeric_parameter

Each Block_base_shape has the width defined by exactly one Numeric_parameter. Each Numeric_parameter defines the width for zero, one, or many Block_base_shape objects.

Each Block_base_shape has the height defined by exactly one Numeric_parameter. Each Numeric_parameter defines the height for zero, one, or many Block_base_shape objects.

 

### 4.3.14 Class_BSU to Supplier_BSU

Each Class_BSU has the defined_by defined by exactly one Supplier_BSU. Each Supplier_BSU defines defined_by for zero, one, or many Class_BSU objects.

### 4.3.15 Company_contract_assignment to Company

Each Company_contract_assignment has the assigned_company defined by exactly one Company objects. Each Company defines the assigned_company for zero, one, or many Company_contract_assignment objects.

### 4.3.16 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.

### 4.3.17 Contract to Company_contract_assignment

Each Contract has the binding_agreement defined by exactly one Company_contract_assignment objects. Each Company_contract_assignment defines the issues for one, or more Contract objects.

### 4.3.18 Contract to Document_file

Each Contract has the contract_document defined by zero or one Document_file objects. Each Document_-file defines the contract_document for zero, one, or many Contract objects.

### 4.3.19 Contract to Part_version

Each Contract has the acquires defined by exactly one Part_version objects. Each Part_version defines the acquires for zero, one, or many Contract objects.

### 4.3.20 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.

### 4.3.21 Controller_program to Document_file

Each Controller_program has the controller_data_file defined by zero, one, or many Document_file objects. Each Document_file defines the controller_data_file for zero, one, or many Controller_program objects.

### 4.3.22 Controller_program to Part_version

Each Controller_program has the identified_on defined by exactly one Part_version objects. Each Part_-version defines the identified_on for zero, one, or many Controller_program objects.

### 4.3.23 Controller_program to Manufacturing_activity

Each Controller_program has the accomplished_by defined by zero, one, or many Manufacturing_activity objects. Each Manufacturing_activity defines the accomplished_by for zero, one, or many Controller_-program objects.

### 4.3.24 Controller_program to to Manufacturing_process

Each Controller_program has the covers defined by zero, one, or many Manufacturing_process objects. Each Manufacturing_process defines the covers for zero, one, or many Controller_program objects.

### 4.3.25 Controller_program to Tool_assembly

Each Controller_program has the references defined by zero, one, or many Tool_assembly objects. Each Tool_assembly defines the references for zero, one, or many Controller_program objects.

### 4.3.26 Cylindrical_base_shape to Numeric_parameter

Each Cylindrical_base_shape has the diameter defined by exactly one Numeric_parameter. Each Numeric_parameter defines the diameter for zero, one, or many Cylindrical_base_shape objects.

### 4.3.27 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.

### 4.3.28 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.

### 4.3.29 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.

### 4.3.30 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.

 

### 4.3.31 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.

### 4.3.32 Distance_along_curve_tolerance to Shape_aspect

Each Distance_along_curve_tolerance has the path defined by exactly one Shape_aspect. Each Shape_-aspect defines the path for zero, one, or many Distance_along_curve_tolerance objects.

### 4.3.33 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.

### 4.3.34 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.

### 4.3.35 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.

### 4.3.36 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.

### 4.3.37 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.

### 4.3.38 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.

### 4.3.39 Explicit_base_shape_representation to Shape_aspect_representation

Each Explicit_base_shape_representation has the shape_form defined by exactly one Shape_aspect_-representation. Each Shape_aspect_representation defines the shape_form for zero, one, or many Explicit_-base_shape_representation objects.

### 4.3.40 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.

### 4.3.41 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_representatiion objects.

### 4.3.42 Externally_defined_size_dimension to Shape_aspect

Each Externally_defined_size_dimension has the path defined by zero or one Shape_aspect. Each Shape_-aspect defines the path for zero, one, or many Externally_defined_size_dimension objects.

### 4.3.43 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.

### 4.3.44 Feature_dependency to Feature_identification_item

Each Feature_dependency has the related_feature_item defined by exactly one Feature_identification_item objects. Each Feature_identification_item defines the related_feature_item for zero, one, or many Feature_dependency objects.

Each Feature_dependency has the relating_feature_item defined by exactly one Feature_identification_-item objects. Each Feature_identification_item defines the relating_feature_item for zero, one, or many Feature_dependency objects.

### 4.3.45 Feature_identification_item to Feature_interaction

Each Feature_identification_item has the feature_relation defined by zero, one, or many Feature_-interaction objects. Each Feature_interaction defines the feature_relation for zero, one, or many Feature_-identification_item objects.

 

### 4.3.46 Feature_identification_item to Manufacturing_process_feature

Each Feature_identification_item has the feature defined by exactly one Manufacturing_process_feature objects. Each Manufacturing_process_feature defines the feature for zero, one, or many Feature_-identification_item objects.

### 4.3.47 Feature_identification_item to Feature_process

Each Feature_identification_item has the process defined by zero, one, or many Feature_process objects. Each Feature_process defines the process for zero, one, or many Feature_identification_item objects.

### 4.3.48 Feature_interaction to Feature_identification_item

Each Feature_interaction has the related_feature defined by exactly one Feature_identification_item objects. Each Feature_identification_item defines the related_feature for zero, one, or many Feature_-interaction objects.

### 4.3.49 Feature_process to Manufacturing_activity

Each Feature_process has the assigned_operation defined by one or more Manufacturing_activity objects. Each Manufacturing_activity defines the assigned_operation for zero, one, or many Feature_process objects.

### 4.3.50 Feature_process to Property_parameter

Each Feature_process has the parameter_for_process defined by zero, one or many Property_parameter objects. Each Property_parameter defines the parameter_for_process for zero, one, or many Feature_-process objects.

### 4.3.51 Feature_process to Manufacturing_process

Each Feature_process has the assigned_process defined by exactly one Manufacturing_process objects. Each Manufacturing_process defines the assigned_process for zero, one, or many Feature_process objects.

### 4.3.52 Fixture_assembly to Design_reference

Each Fixture_assembly has the documented_by defined by zero or one Design_reference objects. Each Design_reference defines the documented_by for zero, one, or many Fixture_assembly objects.

### 4.3.53 Fixture_assembly to Fixture_assembly

Each Fixture_assembly has the composed_of defined by one or more Fixture_assembly objects. Each Fixture_assembly defines the composed_of for zero or one Fixture_assembly objects.

NOTE This assertion is established through the Sub_assembly_select.

### 4.3.54 Fixture_assembly to Fixture_assembly_element

Each Fixture_assembly has the composed_of defined by one or more Fixture_assembly_element objects. Each Fixture_assembly_element defines the composed_of for exactly one Fixture_assembly objects.

NOTE This assertion is established through the Sub_assembly_select.

### 4.3.55 Fixture_assembly to Fixture_contract

Each Fixture_assembly has the identified_by defined by one or more Fixture_contract objects. Each Fixture_contract defines the identifies for exactly one Fixture_assembly objects.

### 4.3.56 Fixture_assembly to Shape_aspect

Each Fixture_assembly has the defined_shape defined by zero, one or many Shape_aspect objects. Each Shape_aspect defines the defined_shape for zero, one, or many Fixture_assembly objects.

### 4.3.57 Fixture_assembly_element to Shape_aspect

Each Fixture_assembly_element has the defined_shape defined by zero, one or many Shape_aspect objects. Each Shape_aspect defines the defined_shape for zero, one or many Fixture_assembly_element objects.

### 4.3.58 Fixture_contract to In_facility_location

Each Fixture_contract has the fixture_location defined by zero or one In_facility_location. Each In_facility_location locates the fixture_location for zero, one, or many Fixture_contract objects.

### 4.3.59 Fixture_pallet_mounting to Machine

Each Fixture_pallet_mounting has the identifies defined by exactly zero Machine objects. Each Machine defines the identifies for zero, one, or many Fixture_pallet_mounting objects.

### 4.3.60 Fixture_pallet_unmounting to Machine

Each Fixture_pallet_unmounting has the identifies defined by exactly zero Machine objects. Each Machine defines the identifies for zero, one, or many Fixture_pallet_unmounting objects.

### 4.3.61 Fixture_setup to Machine

Each Fixture_setup has the identifies defined by exactly one Machine objects. Each Machine defines the identifies for zero, one, or many Fixture_setup objects.

### 4.3.62 Fixture_setup to Mounting_position

Each Fixture_setup has the is_positioned_on defined by zero or one Mounting_position objects. Each Mounting_position defines the is_positioned_on for zero, one, or many Fixture_setup objects.

 

### 4.3.63 Fixture_setup to Fixture_assembly

Each Fixture_setup has the identifies_fixture defined by one or more Fixture_assembly objects. Each Fixture_assembly defines the identifies_fixture for zero, one, or many Fixture_setup objects.

### 4.3.64 Fixture_setup to Pallet

Each Fixture_setup has the identifies_pallet defined by zero, one, or many Pallet objects. Each Pallet defines the identifies_pallet for zero, one, or many Fixture_setup objects.

### 4.3.65 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.

### 4.3.66 Height_dimension to Shape_aspect

Each Height_dimension has the path defined by zero or one Shape_aspect. Each Shape_aspect defines the path for zero, one, or many Height_dimension objects.

### 4.3.67 Illustration to Activity

Each Illustration has the clarifies defined by zero, one or many Activity objects. Each Activity defines the clarifies for zero, one, or many Illustration objects.

### 4.3.68 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.

### 4.3.69 Implicit_base_shape_representation to Numeric_parameter

Each Implicit_base_shape_representation has the base_shape_length defined by exactly one Numeric_-parameter. Each Numeric_parameter defines the base_shape_length for zero, one, or many Implicit_base_-shape_representation objects.

### 4.3.70 Implicit_base_shape_representation to Orientation

Each Implicit_base_shape_representation has the placement defined by exactly one Orientation. Each Orientation defines the placement for zero, one, or many Implicit_base_shape_representation objects.

### 4.3.71 In_facility_location to Work_cell

Each In_facility_location has the locates defined by exactly one Work_cell. Each Work_cell locates the locates for zero, one, or many In_facility_location objects.

### 4.3.72 Intermediate_shape to Shape_aspect

Each Intermediate_shape has the as_is_shape defined by exactly one Shape_aspect. Each Shape_aspect

defines the as_is_shape for zero, one, or many Intermediate_shape objects.

Each Intermediate_shape has the to_be_shape defined by exactly one Shape_aspect. Each Shape_aspect defines the to_be_shape for zero, one, or many Intermediate_shape objects.

### 4.3.73 Length_dimension to Shape_aspect

Each Length_dimension has the path defined by zero or one Shape_aspect. Each Shape_aspect defines the path for zero, one, or many Length_dimension objects.

### 4.3.74 Library_part_assignment to Class_BSU

Each Library_part_assignment has the 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.

### 4.3.75 Library_part_assignment to Property_value

Each Library_part_assignment has the 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.

### 4.3.76 Location_dimension_tolerance to Orientation

Each Location_dimension_tolerance has the plane_and_direction defined by zero or one Orientation. Each Orientation defines the plane_and_direction for zero, one, or many Location_dimension_tolerance objects.

### 4.3.77 Location_tolerance to Shape_aspect

Each Location_tolerance has the termination_shape defined by exactly one Shape_aspect. Each Shape_-aspect defines the termination_shape for zero, one, or many Location_tolerance objects.

Each Location_tolerance has the origin_shape defined by exactly one Shape_aspect. Each Shape_aspect defines the origin_shape for zero, one, or many Location_tolerance objects.

### 4.3.78 Machine to Controller

Each Machine has the controlled_by defined by zero or one Controller. Each Controller defines the controlled_by for zero, one, or many Machine objects.

### 4.3.79 Machine to Pallet

Each Machine has the contains defined by zero, one, or many Pallet objects. Each Pallet defines the contains for zero, one, or many Machine objects.

### 4.3.80 Machine to Workstation

Each Machine has the possessed_by defined by exactly one Workstation. Each Workstation defines the possessed_by for zero, one, or many Machine objects.

 

### 4.3.81 Machine_parameters to Discriptive_parameter

Each Machine_parameters has the axis_range_of_motion_description defined by zero or two or more Descriptive_parameter. Each Descriptive_parameter defines the axis_range_of_motion_description for zero, one, or many Machine_parameters objects.

### 4.3.82 Machine_parameters to Numeric_parameter

Each Machine_parameters has the axis_range_of_motion defined by zero or two or more Numeric_-parameter. Each Numeric_parameter defines the axis_range_of_motion for zero, one, or many Machine_ parameters objects.

Each Machine_parameters has the maximum_feedrate defined by zero or one Numeric_parameter. Each Numeric_parameter defines the maximum_feedrate for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the maximum_spindle_speed defined by zero or one Numeric_parameter. Each Numeric_parameter defines the maximum_spindle_speed for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the number_of_control_axis defined by zero or one Numeric_parameter. Each Numeric_parameter defines the number_of_control_axis for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the number_of_simultaneous_axis defined by zero or one Numeric_-parameter. Each Numeric_parameter defines the number_of_simultaneous_axis for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the positioning_accuracy defined by zero or one Numeric_parameter. Each Numeric_parameter defines the positioning_accuracy for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the spindle_power defined by zero or one Numeric_parameter. Each Numeric_parameter defines the spindle_power for zero, one, or many Machine_parameters objects.

Each Machine_parameters has the table_size defined by zero or one or two Numeric_parameter. Each Numeric_parameter defines the table_size for zero, one, or many Machine_parameters objects.

### 4.3.83 Machine_parameters to Property_parameter

Each Machine_parameters has the machine_parameters defined by zero, one, or many Property_parameter. Each Property_parameter defines the machine_parameters for zero, one, or many Machine_parameters objects.

### 4.3.84 Machine_setup to Machine

Each Machine_setup has the identifies defined by exactly one Machine. Each Machine defines the identifies for zero, one, or many Machine_setup objects.

### 4.3.85 Machine_usage to Machine

Each Machine_usage has the classifies defined by exactly one Machine. Each Machine defines the classifies for zero, one, or many Machine_usage objects.

### 4.3.86 Machine_usage to Machine_parameters

Each Machine_usage has the machine_parameter defined by zero or one Machine_parameters. Each Machine_parameters defines the machine_parameter for zero, one, or many Machine_usage objects.

### 4.3.87 Machine_usage to Specification

Each Machine_usage has the machine_specification defined by zero or one Specification. Each Specification defines the machine_specification for zero, one, or many Machine_usage objects.

### 4.3.88 Machining_process to Machine

Each Machining_process has the required_machine defined by exactly one Machine. Each Machine defines the required_machine for zero, one, or many Machining_process objects.

### 4.3.89 Manufactured_assembly_relationship to Manufactured_assembly

Each Manufactured_assembly_relationship has the assembly defined by exactly one Manufactured_-assembly. Each Manufactured_assembly has components defined by two or more Manufactured_-assembly_relationship objects.

### 4.3.90 Manufactured_assembly_relationship to Part_placement

Each Manufactured_assembly_relationship has the orientation defined by exactly one Part_placement. Each Part_placement defines the orientation for zero, one, or many Manufactured_assembly_relationship objects.

### 4.3.91 Manufactured_assembly_relationship to Part_version

Each Manufactured_assembly_relationship has the component defined by exactly one Part_version. Each Part_version defines the component for zero, one, or many Manufactured_assembly_relationship objects.

### 4.3.92 Manufacturing_activity to External_schema_definition

Each Manufacturing_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 Manufacturing_activity objects.

### 4.3.93 Manufacturing_process to Intermediate_shape

Each Manufacturing_process has the in_process_shape defined by zero or one Intermediate_shape objects. Each Intermediate_shape defines the in_process_shape for exactly zero, one, or many Manufacturing_-process.

 

### 4.3.94 Manufacturing_process to Manufacturing_activity

Each Manufacturing_process has the assigned_operation defined by one or more Manufacturing_activity. Each Manufacturing_activity defines the assigned_operation for zero, one, or many Manufacturing_-process objects.

### 4.3.95 Manufacturing_process to Manufacturing_process_feature

Each Manufacturing_process has the assigned_feature defined by zero, one, or many Manufacturing_-process_feature objects. Each Manufacturing_process_feature defines the assigned_feature for zero, one, or many Manufacturing_process.

### 4.3.96 Manufacturing_process to Part_holding_position

Each Manufacturing_process has the part_hold_downs defined by zero, one or many Part_holding_-position objects. Each Part_holding_position defines the part_hold_downs for one or more Manufacturing_process objects.

### 4.3.97 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 one or more Manufacturing_process objects.

### 4.3.98 Material to Design_reference

Each Material has the identified_by_drawing defined by zero or one Design_reference. Each Design_-reference defines the identified_by_drawing for zero, one, or many Material objects.

### 4.3.99 Material to Material_specification

Each Material has the documented_by defined by zero or one Material_specification. Each Material_-specification defines the documented_by for zero, one, or many Material objects.

### 4.3.100 Material to Base_shape

Each Material has the initial_material_shape defined by exactly one Base_shape. Each Base_shape defines the initial_material_shape for zero, one, or many Material objects.

NOTE This assertion is established through the base_shape select type.

### 4.3.101 Material to Material_property

Each Material has the material_characteristics defined by zero, one, or many Material_property. Each Material_property defines the material_characteristics for zero, one, or many Material objects.

### 4.3.102 Material_property to Hardness

Each Material_property has the material_hardness defined by zero, one, or many Hardness objects. Each Hardness defines the material_hardness for zero, one, or many Material_property.

### 4.3.103 Material_property to Property_parameter

Each Material_property has the material_parameters defined by zero, one, or many Property_parameter objects. Each Property_parameter defines the material_parameters for zero, one, or many Material_-property.

### 4.3.104 Mating_definition to Manufactured_assembly

Each Mating_definition has the applied_assembly defined by exactly one Manufactured_assembly. Each Manufactured_assembly defines the applied_assembly for zero, one, or many Mating_definition objects.

### 4.3.105 Mating_definition to Shape_aspect

Each Mating_definition has the mating_shape defined by two, or more Shape_aspect objects. Each Shape_aspect object defines the mating_shape for zero, one, or many Mating_definition objects.

### 4.3.106 Mating_definition to Single_piece_part

Each Mating_definition has the mating_solution defined by zero, one, or many Single_piece_part objects. Each Single_piece_part object defines the mating_solution for zero, one, or many Mating_definition objects.

### 4.3.107 Mating_definition_relationship to Mating_definition

Each Mating_definition_relationship has the mating_part_definiton defined by exactly one Mating_-definition. Each Mating_definition object defines the mated_part for two, or more Mating_definition_-relationship objects.

### 4.3.108 Mating_definition_relationship to Part_placement

Each Mating_definition_relationship has the orientation defined by zero or one Part_placement object. Each Part_placement object defines the orientation for zero, one, or many Mating_definition_relationship objects.

### 4.3.109 Mating_definition_relationship to Single_piece_part

Each Mating_definition_relationship has the mated_part defined by exactly one Single_piece_part. Each Single_piece_part object defines the mated_part for zero, one, or many Mating_definition_relationship objects.

 

### 4.3.110 Mating_relationship to Single_piece_part

Each Mating_relationship has the predecessor defined by exactly one Single_piece_part. Each Single_-piece_part object defines the predecessor for zero, one, or many Mating_relationship objects.

Each Mating_relationship has the successor defined by exactly one single_piece_part. Each Single_piece_-part object defines the successor for zero, one, or many Mating_relationship objects.

### 4.3.111 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.

### 4.3.112 Model_element to Shape_representation_item

Each Model_element defines the representation_type by exactly one Shape_representation_item. Each Shape_representation_item defines the representation_type for zero, one, or many Model_element objects.

### 4.3.113 Ngon_base_shape to Numeric_parameter

Each Ngon_base_shape has the number_of_sides defined by exactly one Numeric_parameter. Each Numeric_parameter defines the number_of_sides for zero, one, or many Ngon_base_shape objects.

Each Ngon_base_shape has the diameter defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the diameter for zero, one, or many Ngon_base_shape objects.

Each Ngon_base_shape has the corner_radius defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the corner_radius for zero, one, or many Ngon_base_shape objects.

### 4.3.114 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.

### 4.3.115 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.

### 4.3.116 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.

### 4.3.117 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.

### 4.3.118 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.

### 4.3.119 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.

### 4.3.120 Pallet_machine_mounting to Pallet

Each Pallet_machine_mounting has the identifies_pallet defined by exactly one Pallet objects. Each Pallet defines the identifies_pallet for zero, one or many Pallet_machine_mounting objects.

### 4.3.121 Pallet_machine_mounting to Machine

Each Pallet_machine_mounting has the identifies_machine defined by exactly one Machine objects. Each Machine defines the identifies_machine for zero, one or many Pallet_machine_mounting objects.

### 4.3.122 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.

### 4.3.123 Part_fixture_mounting to Fixture_assembly

Each Part_fixture_mounting has the identifies_fixture defined by one or more Fixture_assembly objects. Each Fixture_assembly defines the identifies_fixture for zero, one or many Part_fixture_mounting objects.

 

### 4.3.124 Part_holding_position to Fixture_assembly

Each Part_holding_position has the part_hold_downs by one or more Fixture_assembly objects. Each Fixture_assembly defines the part_hold_downs for zero, one, or many Part_holding_position objects.

### 4.3.125 Part_holding_position to Shape_aspect

Each Part_holding_position has the mating_shape defined by zero, one or more Shape_aspect objects. Each Shape_aspect defines the mating_shape for zero, one, or many Part_holding_position objects.

### 4.3.126 Part_machine_mounting to Machine

Each Part_machine_mounting has the identifies_machine defined by exactly one Machine. Each Machine defines the identifies_machine for zero, one, or many Part_machine_mounting objects.

### 4.3.127 Part_mounting to Mounting_position

Each Part_mounting has the is_positioned_on defined by zero or one Mounting_position objects. Each Mounting_position defines the is_positioned_on for zero, one, or many Part_mounting objects.

### 4.3.128 Part_mounting to Part_shape

Each Part_mounting has the is_located_on defined by zero or one Part_shape objects. Each Part_shape defines the is_located_on for zero, one or many Part_mounting objects.

### 4.3.129 Part_placement to Orientation

Each Part_placement has the resulting_orientation defined by exactly one Orientation. Each Orientation defines the resulting_orientation for zero, one, or many Part_placement objects.

Each Part_placement has the originating_orientation defined by exactly one Orientation. Each Orientation defines the originating_orientation for zero, one, or many Part_placement objects.

### 4.3.130 Part_placement to Part_shape

Each Part_placement has the oriented_physical_form defined by exactly one Part_shape. Each Part_shape defines the oriented_physical_form for zero, one, or many Part_placement objects.

### 4.3.131 Part_property to Property_parameter

Each Part_property has the part_characteristics defined by zero, one, or many Property_parameter objects. Each Property_parameter defines the part_characteristics for zero, one, or many Part_property.

### 4.3.132 Part_shape to Base_shape

Each Part_shape has the base_shape_definition defined by zero or one Base_shape. Each Base_shape defines the base_shape_definition for exactly one Part_shape.

### 4.3.133 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.

### 4.3.134 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.

### 4.3.135 Part_version to Alternate_process_plan

Each Part_version has the alternate_plan defined by zero, one or many Alternate_process_plan objects. Each Alternate_process_plan defines the alternate_plan for exactly one Part_version objects.

### 4.3.136 Part_version to Design_reference

Each Part_version has the documented_by defined for zero or one Design_reference objects. Each Design_reference defines the documents_version for zero, one, or many Part_version objects.

### 4.3.137 Part_version to Part_shape

Each Part_version has the physical_form defined by exactly one Part_shape objects. Each Part_shape defines the physical_form for zero, one or many Part_version objects.

### 4.3.138 Part_version to Process_plan_version

Each Part_version has the fabrication_defined_by defined by exactly one Process_plan_version objects. Each Process_plan_version defines the defines_fabrication_for for exactly one Part_version objects.

### 4.3.139 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 objects.

### 4.3.140 Planning_group_member to Organization

Each Planning_group_member has the employed_by defined by exactly one Organization objects. Each Organization defines the employed_by for zero, one or many Planning_group_member objects.

### 4.3.141 Process_activity to Fixture_assembly

Each Process_activity has the identifies_fixture defined by zero or one Fixture_assembly objects. Each Fixture_assembly defines the identifies_fixture for zero, one or many Process_activity objects.

 

### 4.3.142 Process_activity to Machine_setup

Each Process_activity has the identifies defined by zero or one Machine_setup objects. Each Machine_-setup defines the identifies for zero, one or many Process_activity objects.

### 4.3.143 Process_activity to Machining_tolerance

Each Process_activity defines the tolerances for zero or one Machining_tolerance objects. Each Machining_tolerance has the specified_by defined by exactly one Process_activity.

### 4.3.144 Process_activity to Process_property

Each Process_activity has the process_parameters defined by zero, one or many Process_property objects. Each Process_property defines the process_parameters for zero, one or many Process_activity objects.

### 4.3.145 Process_activity to Shape_aspect

Each Process_activity has the described_by defined by zero or one Shape_aspect objects. Each Shape_-aspect defines the described_by for zero, one or many Process_activity objects.

### 4.3.146 Process_activity to Tool_assembly

Each Process_activity has the uses_to_perform defined by zero or one Tool_assembly objects. Each Tool_-assembly defines the uses_to_perform for zero, one or many Process_activity objects.

### 4.3.147 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.

### 4.3.148 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.

### 4.3.149 Process_plan_version to Feature_dependency

Each Process_plan_version has the feature_dependency_suggestion defined by zero, one or many Feature_dependency objects. Each Feature_dependency defines the feature_dependency_suggestion for zero, one or many Process_plan_version objects.

### 4.3.150 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.

### 4.3.151 Process_plan_version to Material

Each Process_plan_version has the required_material defined by exactly one Material objects. Each Material defines the identified_by for zero, one or many Process_plan_version objects.

### 4.3.152 Process_plan_version to Property_parameter

Each Process_plan_version has the auxiliary_header_information defined by zero, one or many Property_-parameter objects. Each Property_parameter defines the auxiliary_header_information for zero, one or many Process_plan_version objects.

### 4.3.153 Process_plan_version to Range_of_parts

Each Process_plan_version has the quantity_range defined by zero or one Range_of_parts objects. Each Range_of_parts defines the quantity_range for zero, one or many Process_plan_version objects.

### 4.3.154 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.

### 4.3.155 Process_property to Property_parameter

Each Process_property has the process_characteristics defined by zero, one, or many Property_parameters objects. Each Property_parameters defines the process_characteristics for zero, one, or many Process_-property.

### 4.3.156 Property to Material_property

Each Property has the material_characteristics defined by zero, one, or many Material_property objects. Each Material_property defines the material_characteristics for zero, one, or many Property objects.

### 4.3.157 Property to Part_property

Each Property has the part_property_characteristics defined by zero, one, or many Part_property objects. Each Part_property defines the part_property_characteristics for one or more Property objects.

### 4.3.158 Property to Process_property

Each Property has the process_characteristics defined by zero, one, or many Process_property objects. Each Process_property defines the process_characteristics for one or more Property objects.

### 4.3.159 Property to Shape_aspect

Each Property has the property_characteristiscs defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the property_characteristics for zero, one, or many Property objects.

 

### 4.3.160 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.

### 4.3.161 Property to Surface_property

Each Property has the surface_characteristics defined by zero, one, or many Surface_property objects. Each Surface_property defines the surface_characteristics for one or more Property objects.

### 4.3.162 Property_BSU to Class_BSU

Each Property_BSU has the name_scope defined by exactly one Class_BSU. Each Class_BSU defines the name_scope for zero, one, or many Property_BSU objects.

### 4.3.163 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.

### 4.3.164 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.

### 4.3.165 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.

### 4.3.166 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.

### 4.3.167 Revision to Status_authority

Each Revision has the approved_by defined by one or more Status_authority objects. Each Status_-authority defines the approved_by for zero, one, or many Revision objects.

### 4.3.168 Shape_aspect to Manufacturing_process_feature

Each Shape_aspect has the element defined by zero or one Manufacturing_process_feature. Each Manufacturing_process_feature defines the element for exactly one Shape_aspect.

 

### 4.3.169 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.

### 4.3.170 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.

### 4.3.171 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 zero, one, or many Shape_aspect_representation.

### 4.3.172 Single_piece_part to Alternate_material

Each Single_piece_part has the alternate_material_definition defined by zero, one, or many Alternate_-material objects. Each Alternate_material defines the replacement_material for one or more Single_piece_-part.

### 4.3.173 Single_piece_part to Material

Each Single_piece_part has the material_definition defined by one or more Material objects. Each Material defines the material_definition for one or more Single_piece_part objects.

### 4.3.174 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_tolerance objects.

### 4.3.175 Special_capability to Workstation

Each Special_capability has the possessed_by defined by exactly one Workstation. Each Workstation defines the possessed_by for zero, one, or many Special_capability objects.

### 4.3.176 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.

### 4.3.177 Status_authority to Planning_group_member

Each Status_authority has the is_given_by defined by exactly one Planning_group_member. Each Planning_group_member defines the is_given_by for zero, one, or many Status_authority objects.

### 4.3.178 Surface_property to Property_parameter

Each Surface_property has the surface_characteristics defined by zero, one, or many Property_parameter objects. A Property_parameter defines the surface_characteristics for zero, one, or many Surface_property.

 

### 4.3.179 Thickness_tolerance to Shape_aspect

Each Thickness_tolerance has the path defined by exactly one Shape_aspect. Each Shape_aspect defines the path for zero, one, or many Thickness_tolerance objects.

### 4.3.180 Tolerance_value to Limits_and_fits

Each Tolerance_value has the defined_value defined by exactly one Limit_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.

### 4.3.181 Tolerance_value to Plus_minus_value

Each Tolerance_value has the defined_value defined by exactly 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.

### 4.3.182 Tolerance_value to Tolerance_limit

Each Tolerance_value has the defined_value defined by exactly 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.

### 4.3.183 Tolerance_value to Tolerance_range

Each Tolerance_value has the defined_value defined by exactly 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.

### 4.3.184 Tool_assembly to Design_reference

Each Tool_assembly has the documented_by defined by zero or one Design_reference objects. Each Design_reference defines the documented_by for zero, one, or many Tool_assembly objects.

### 4.3.185 Tool_assembly to Property_parameter

Each Tool_assembly has the tool_parameters defined by zero, one, or many Property_parameter. Each Property_parameter defines the tool_parameters for zero, one, or many Tool_assembly objects.

### 4.3.186 Tool_assembly to Shape_aspect

Each Tool_assembly has the defined_shape defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the defined_shape for zero, one, or many Tool_assembly objects.

## 4.3.187 Tool_assembly to Tool_assembly_element

Each Tool_assembly has the composed_of defined by one or more Tool_assembly_element objects. Each Tool_assembly_element defines the used_in for one or more Tool_assembly objects.

## 4.3.188 Tool_assembly to Tool_contract

Each Tool_assembly has the identified_in_contract defined by zero or one Tool_contract. Each Tool_-contract defines the identified_in_contract for exactly one Tool_assembly objects.

## 4.3.189 Tool_assembly to Tool_placement_instruction

Each Tool_assembly defines the identifying_data for zero, one, or many Tool_placement_instruction objects. Each Tool_placement_instruction has the identifying_data defined by exactly one Tool_assembly.

## 4.3.190 Tool_assembly_element to Externally_defined_representation

Each Tool_assembly_element has the tool_representation defined by zero or one Externally_defined_-representation. Each Externally_defined_representation defines the tool_representation for zero, one, or many Tool_assembly_element objects.

## 4.3.191 Tool_assembly_element to Shape_aspect

Each Tool_assembly_element has the defined_shape defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the defined_shape for zero, one, or many Tool_assembly_element

## 4.3.192 Tool_assembly_element to Specification

Each Tool_assembly_element has the tool_property defined by zero, one, or many Specification. Each Specification defines the tool_property for zero, one, or many Tool_assembly_element objects.

## 4.3.193 Tool_body to Property_parameter

Each Tool_body has the tool_body_parameter defined by zero, one, or many Property_parameter. Each Property_parameter defines the tool_body_parameter for zero, one, or many Tool_body objects.

## 4.3.194 Tool_contract to Specification

Each Tool_contract has the contract_document defined by zero or one Specification. Each Specification defines the contract_document for zero, one, or many Tool_contract objects.

## 4.3.195 Tool_placement_instruction to Specification

Each Tool_placement_instruction has the instruction_document defined by zero or one Specification. Each Specification defines the instruction_document for zero, one, or many Tool_placement_instruction objects.

 

### 4.3.196 Tool_placement_instruction to Tool_magazine_turret_carousel

Each Tool_placement_instruction has the configures defined by exactly one Tool_magazine_turret_-carousel. Each Tool_magazine_turret_carousel defines the configures for zero, one, or many Tool_-placement_instruction objects.

### 4.3.197 Tool_magazine_turret_carousel to Workstation

Each Tool_magazine_turret_carousel has the used_in defined by exactly one Workstation. Each Workstation defines the used_in for zero, one, or many Tool_magazine_turret_carousel objects.

### 4.3.198 Tool_setup to Machine

Each Tool_setup has the identifies_machine defined by exactly one Machine. Each Machine defines the identifies_machine for zero, one, or many Tool_setup objects.

### 4.3.199 Tool_setup to Tool_assembly

Each Tool_setup has the identifies_tool defined by one or more Tool_assembly. Each Tool_assembly defines the identifies_tool for zero, one, or many Tool_setup objects.

### 4.3.200 Tool_setup to Tool_magazine_turret_carousel

Each Tool_setup has the tool_placement defined by exactly one Tool_magazine_turret_carousel. Each Tool_magazine_turret_carousel defines the tool_placement for zero, one, or many Tool_setup objects.

### 4.3.201 Validation to Controller_program

Each Validation has the checks defined by zero or one Controller_program. Each Controller_program defines the checks for zero, one, or many Validation objects.

### 4.3.202 Validation to Fixture_assembly

Each Validation has the identifies defined by zero or one Fixture_assembly. Each Fixture_assembly defines the identifies for zero, one, or many Validation objects.

### 4.3.203 Validation to Part_shape

Each Validation has the uses_for_validation defined by exactly one Part_shape. Each Part_shape defines the uses_for_validation for zero, one, or many Validation objects.

### 4.3.204 Validation to Tool_assembly

Each Validation has the uses_to_perform defined by exactly one Tool_assembly. Each Tool_assembly defines the uses_to_perform for zero, one, or many Validation objects.

### 4.3.205 View_reference to Activity

Each View_reference has the identifies defined by zero, one, or many Activity. Each Activity defines the identifies for zero, one, or many View_reference objects.

### 4.3.206 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.

### 4.3.207 Width_dimension to Shape_aspect

Each Width_dimension has the path defined by zero or one Shape_aspect. Each Shape_aspect defines the path for zero, one, or many Width_dimension objects.

### 4.3.208 Work_cell to Activity

Each Work_cell has the referenced_by defined by zero, one, or many Activity. Each Activity defines the work_cell_reference for exactly one Work_cell objects.

### 4.3.209 Work_cell to Workstation

Each Work_cell has the contains defined by one or more Workstation. Each Workstation defines the contained_in for exactly one Work_cell 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 administrative UoF

#### 5.1.1.1 Company

AIM element: organization
Source: ISO 10303-41

##### 5.1.1.1.1 address

AIM element: organizational_address
Source: ISO 10303-41
Reference path: organization <-
organizational_address.organizations[i]
organizational_address

##### 5.1.1.1.2 name

AIM element: organization.name
Source: ISO 10303-41

#### 5.1.1.2 Company_contract_assignment

AIM element: applied_organization_assignment
Source: ISO 10303-240
Reference path: applied_organization_assignment < =
organization_assignment

##### 5.1.1.2.1 role

AIM element: organization_role
Source: ISO 10303-41
Reference path: applied_organization_assignment < =
organization_assignment
organization_assignment.role - >
organization_role

##### 5.1.1.2.2 company_contract_assignment to company (as assigned_company)

AIM element: PATH
Reference path: applied_organization_assignment < =
organization_assignment
organization_assignment.assigned_organization - >
organization

#### 5.1.1.3 Organization

AIM element: organization
Source: ISO 10303-41

 

#### 5.1.1.3.1 address_of_site

AIM element: organizational_address
Source: ISO 10303-41
Reference path: organization <-
organizational_address.organizations [i]
organizational_address

#### 5.1.1.3.2 name

AIM element: organization.name
Source: ISO 10303-41

### 5.1.1.4 Planning_group_member

AIM element: person
Source: ISO 10303-41

#### 5.1.1.4.1 department

AIM element: organization.id
Source: ISO 10303-41
Reference path: person <-
person_and_organization.the_person
person_and_organization
person_and_organization.the_organization ->
organization
organization.id

#### 5.1.1.4.2 id

AIM element: person.id
Source: ISO 10303-41

#### 5.1.1.4.3 name

AIM element: [person.last_name]
[(person.first_name)
(person.middle_name)
(person.prefix_title)
(person.suffix_title)]
Source: ISO 10303-41

### 5.1.1.4.4 phone_number

AIM element: address.telephone_number
Source: ISO 10303-41
Reference path: person <-
personal_address.people [i]
personal_address <=
address
address.telephone_number

### 5.1.1.4.5 planning_group_member to organization (as employed_by)

AIM element: PATH
Source: ISO 10303-41
Reference path: person <-
person_and_organization.the_person
person_and_organization
person_and_organization.the_organization ->
organization

### 5.1.1.5 Status_authority

AIM element: approval
Source: ISO 10303-41
Rule: 5.2.3.4.2,
5.2.3.4.3

### 5.1.1.5.1 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

 

#### 5.1.1.5.2 date

AIM element: calendar_date
Source: ISO 10303-41
Rules: 5.2.3.4.2
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

#### 5.1.1.5.3 status_authority to planning_group_member (as is_given_by)

AIM element: PATH
Rules: 5.2.3.4.3
Reference path: approval<-
approval_person_organization.authorized_approval
approval_person_organization
approval_person_organization.person_organization->
person_organization_select
person_organization_select = person
person

### 5.1.2 design_exception UoF

#### 5.1.2.1 Design_exception_notice

AIM element: versioned_action_request
Source: ISO 10303-41

#### 5.1.2.1.1 issuing_date

AIM element: date
Source: ISO 10303-41
Rules: 5.2.3.4.2
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

#### 5.1.2.1.2 notice_description

AIM element: versioned_action_request.purpose
Source: ISO 10303-41

#### 5.1.2.1.3 notice_number

AIM element: versioned_action_request.id
Source: ISO 10303-41

#### 5.1.2.1.4 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

 

#### 5.1.2.1.5 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'}

#### 5.1.2.1.6 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

### 5.1.2.2 Engineering_change_order

AIM element: action_directive
Source: ISO 10303-41

#### 5.1.2.2.1 change_order_number

AIM element: action_directive.name
Source: ISO 10303-41

### 5.1.2.2.2 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

## 5.1.2.3 Engineering_change_proposal

AIM element: document
Source: ISO 10303-41
Reference path: {document
document.kind ->
document_type
document_type.product_data_type = 'engineering change proposal'

### 5.1.2.3.1 change_proposal_number

AIM element: document.id
Source: ISO 10303-41

### 5.1.2.3.2 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 library_reference UoF

#### 5.1.3.1 BSU

#1: BSU is a Class_BSU

AIM element: externally_defined_class
Source: ISO 10303-240

#2: BSU is a Property_BSU

AIM element: externally_defined_general_property
Source: ISO 10303-240

#3: BSU is a Supplier_BSU

AIM element: organization
Source: ISO 10303-41

##### 5.1.3.1.1 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

### 5.1.3.2 Class_BSU

AIM element: externally_defined_class
Source: ISO 10303-240
Rules: 5.2.3.5.2
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'}]

#### 5.1.3.2.1 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

 

#### 5.1.3.2.2 class_BSU to supplier_BSU (as defined_by)

AIM element: organization
Source: ISO 10303-41
Rule: 5.2.3.4.12
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

### 5.1.3.3 Externally_defined_representation

AIM element: externally_defined_representation_with_parameters
Source: ISO 10303-240
Reference path: externally_defined_representation_with_parameters<=
representation

#### 5.1.3.3.1 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

### 5.1.3.3.2 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

### 5.1.3.3.3 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

### 5.1.3.4 Library_part_assignment

AIM element: applied_classification_assignment
Source: ISO 10303-240
Reference path: applied_classification_assignment<=
classification_assignment

### 5.1.3.4.1 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

 

#### 5.1.3.4.2 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'}

### 5.1.3.5 Property_BSU

AIM element: externally_defined_general_property
Source: ISO 10303-240
Rules: 5.2.3.5.2
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'}]

### 5.1.3.5.1 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

### 5.1.3.5.2 property_BSU to class_BSU (as name_scope)

AIM element: PATH
Reference path: externally_defined_general_property <=
externally_defined_item <-
externally_defined_item_relationship.relating_item
externally_defined_item_relationship
{externally_defined_item_relationship.name = 'name scope'}
externally_defined_item_relationship.related_item ->
externally_defined_item =>
externally_defined_class

### 5.1.3.6 Property_value

AIM element: representation
Source: ISO 10303-43
Reference path: representation
{representation.name = 'property value'}

### 5.1.3.6.1 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

 

### 5.1.3.6.2 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

### 5.1.3.7 Supplier_bsu

AIM element: organization
Source: ISO 10303-41
Rules: 5.2.3.5.2, 5.2.3.4.12
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

 

## 5.1.4 manufacturing_machine_tool_resources UoF

### 5.1.4.1 Controller

AIM element: controller
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: controller<=
action_resource

#### 5.1.4.1.1 company_model

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: controller<=
action_resource
action_resource.description

#### 5.1.4.1.2 company_name

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: controller<=
action_resource
action_resource.name

#### 5.1.4.1.3 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

### 5.1.4.2 Fixture_assembly

AIM element: fixture_assembly
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: fixture_assembly<=
action_resource

### 5.1.4.2.1 assembly_instruction

AIM element: action_method.description
Source: ISO 10303-41
Reference path: fixture_assembly<=
action_resource
action_resource.usage ->
supported_item
supported_item=action_method
action_method
action_method.name='assembly instruction'
action_method.description

### 5.1.4.2.2 configuration

AIM element: resource_property.description
Source: ISO 10303-41
Reference path: fixture_assembly<=
action_resource
action_resource=characterized_resource_definition
characterized_resource_definition <-
resource_property.resource
resource_property
resource_property.name='configuration'
resource_property.description

### 5.1.4.2.3 id

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: fixture_assembly<=
action_resource
action_resource.name

### 5.1.4.2.4 fixture_assembly to fixture_assembly_element (as composed_of)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: fixture_assembly<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
fixture_assembly_relationship
action_resource_relationship.related_resource->
action_resource=>
fixture_assembly_element

 

### 5.1.4.2.5 fixture_assembly to fixture_assembly (as composed_of)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: fixture_assembly<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
fixture_assembly_relationship
action_resource_relationship.related_resource->
action_resource=>
fixture_assembly

### 5.1.4.2.6 fixture_assembly to shape_aspect ( as defined_shape)

AIM element: PATH
Reference path: fixture_assembly<=
action_resource <-
requirement_for_action_resource.resources [i]
requirement_for_action_resource <=
action_resource_requirement
action_resource_requirement.operations[i] ->
characterized_action_definition=action
action =>
property_process<-
process_property_association.process
process_property_association
process_property_association.name='fixture assembly shape'
process_property_association.property_or_shape ->
property_or_shape_select=shape_definition
shape_definition=shape_aspect
shape_aspect

### 5.1.4.2.7 fixture_assembly to design_reference (as documented_by)

AIM element: PATH
Reference path: fixture_assembly
document_reference_item=fixture_assembly<-
applied_document_reference.items [i]
applied_document_reference<=
document_reference
document_reference.assigned_document->
document =>
design_reference

### 5.1.4.2.8 fixture_assembly to fixture_contract (as identified_by)

AIM element: PATH
Reference path: fixture_assembly
contract_item=fixture_assembly
contract_item <-
applied_contract_assignment.items [i]
applied_contract_assignment <=
contract_assignment
contract_assignment.assigned_contract ->
contract=>
fixture_contract

### 5.1.4.3 Fixture_assembly_element

AIM element: fixture_assembly_element
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: fixture_assembly_element<=
action_resource

### 5.1.4.3.1 catalogue_number

AIM element: document.id
Source: ISO 10303-41
Reference path: fixture_assembly_element
document_reference_item=fixture_assembly_element
document_reference_item <-
applied_document_reference.items [i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document
{document.name='fixture catalogue number'}
document.id

 

### 5.1.4.3.2 company_name

AIM element: organization.name
Source: ISO 10303-41
Reference path: fixture_assembly_element
organization_item=fixture_assembly_element
organization_item <-
applied_organization_assignment.items [i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role->
organization_role
organization_role.name='fixture company name'}
organization_assignment.assigned_organization ->
organization
organization.name

### 5.1.4.3.3 description

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: fixture_assembly_element<=
action_resource
action_resource.description

### 5.1.4.3.4 id

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: fixture_assembly_element<=
action_resource
action_resource=characterized_resource_definition
characterized_resource_definition <-
resource_property.resource
resource_property
{resource_property.name='fixture element identification'}
resource_property.description

### 5.1.4.3.5 name

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: fixture_assembly_element<=
action_resource
action_resource.name

### 5.1.4.3.6 weight

AIM element: mass_measure_with_unit
Source: ISO 10303-41
Reference path: fixture_assembly_element<=
action_resource
action_resource=characterized_resource_definition
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='fixture element weight'}
measure_representation_item <=
measure_with_unit =>
mass_measure_with_unit

### 5.1.4.3.7 fixture_assembly_element to shape_aspect ( as defined_shape)

AIM element: PATH
Reference path: fixture_assembly_element<=
action_resource <-
requirement_for_action_resource.resources [i]
requirement_for_action_resource <=
action_resource_requirement
action_resource_requirement.operations[i] ->
characterized_action_definition=action
action =>
property_process<-
process_property_association.process
process_property_association
process_property_association.description='fixture assembly element shape'
process_property_association.property_or_shape ->
property_or_shape_select=shape_definition
shape_definition=shape_aspect
shape_aspect

## 5.1.4.4 Generic_manufacturing_resource

AIM element: generic_manufacturing_resource
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: generic_manufacturing_resource<=
action_resource

 

#### 5.1.4.4.1 description

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: generic_manufacturing_resource<=
action_resource
action_resource.description

#### 5.1.4.4.2 name

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: generic_manufacturing_resource<=
action_resource
action_resource.name

#### 5.1.4.4.3 quantity

AIM element: count_measure
Source: ISO 10303-41
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

### 5.1.4.4.4 units

AIM element: context_dependent_unit
Source: ISO 10303-41
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'}

### 5.1.4.5 In_facility_location

AIM element: in_facility_location
Source: ISO 10303-240
Rule: 5.2.3.7.7
Reference path: in_facility_location <=
resource_property

### 5.1.4.5.1 building_or_area

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: in_facility_location<=resource_property
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
{representation.name='building or area'}
representation.items[i] ->
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

 

### 5.1.4.5.2 location_code

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: in_facility_location<=resource_property
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
{representation.name = ‘location code’}
representation.items[i] ->
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

### 5.1.4.5.3 sublocation

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: in_facility_location<=
resource_property
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
{representation.name = ‘sublocation’
representation.items[i] ->
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

### 5.1.4.5.4 in_facility_location to work_cell (as locates)

AIM element: PATH
Reference path: resource_property
resource_property.resource->
characterized_resource_definition
characterized_resource_definition=action_resource
action_resource=>
work_cell

### 5.1.4.6 Machine

AIM element: machine
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: machine<=
action_resource

### 5.1.4.6.1 company_model

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: machine<=
action_resource
action_resource.description

### 5.1.4.6.2 company_name

AIM element: organization.name
Source: ISO 10303-41
Reference path: machine
organization_item=machine
organization_item <-
applied_organization_assignment.items [i]<=
applied_organization_assignment <=
organization_assignment
{organization_assignment.role->
organization_role
organization_role.name='machine company name'}
organization_assignment.assigned_organization ->
organization
organization.name

### 5.1.4.6.3 id

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: machine<=
action_resource
action_resource.name

 

### 5.1.4.6.4 machine to pallet (as contains)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: machine<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
machine_element_relationship
action_resource_relationship.related_resource->
action_resource=>
pallet

### 5.1.4.6.5 machine to controller (as controlled_by)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: machine<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
machine_element_relationship
action_resource_relationship.related_resource->
action_resource=>
controller

### 5.1.4.6.6 machine to workstation (as possessed_by)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: machine<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
machine_element_relationship
action_resource_relationship.related_resource->
action_resource=>
workstation

## 5.1.4.7 Pallet

AIM element: pallet
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: pallet<=
action_resource

### 5.1.4.7.1 identifier

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: pallet<=
action_resource
action_resource.name

### 5.1.4.8 part_holding_position

AIM element: part_holding_position
Source: ISO 10303-240
Rule: 5.2.3.7.7
Reference path: part_holding_position <=
resource_property

### 5.1.4.8.1 location

AIM element: cartesian_point
Source: ISO 10303-42
Reference path: part_holding_position <=
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name =’part holding location’}
geometric_representation_item =>
point =>
cartesian_point

 

### 5.1.4.8.2 position_type

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: part_holding_position <=
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name =’part holding position type’}
descriptive_representation_item
{(descriptive_representation_item.description = ‘jack‘)
(descriptive_representation_item.description = ‘locator‘)
(descriptive_representation_item.description = ‘clamp‘)}

### 5.1.4.8.3 part_holding_position to fixture_assembly (as part_hold_downs)

AIM element: PATH
Reference path: part_holding_position <=
resource_property
resource_property.resource->
characterized_resource_definition=action_resource_requirement
action_resource_requirement=>
requirement_for_action_resource
requirement_for_action_resource.resources[i]->
action_resource=>
fixture_assembly

### 5.1.4.8.4 part_holding_position to shape_aspect (as mating_shape)

AIM element: PATH
Reference path: part_holding_position <=
resource_property
resource_property.resource->
characterized_resource_definition=action_resource_requirement
action_resource_requirement
{action_resource_requirement =>
requirement_for_action_resource }
action_resource_requirement.operations[i] ->
characterized_action_definition=action
{action
action.chosen_method->
action_method=>
manufacturing_process }
action =>
property_process<-
process_property_association.process
process_property_association
{process_property_association.description='part holding mating shape'}
process_property_association.property_or_shape ->
property_or_shape_select=shape_definition
shape_definition=shape_aspect
shape_aspect

### 5.1.4.9 Resource_with_material

AIM element: resource_with_material
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: resource_with_material<=
generic_manufacturing_resource<=
action_resource

 

### 5.1.4.9.1 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

## 5.1.4.10 Resource_with_representation

AIM element: resource_with_representation
Source: ISO 10303-49
Rule: 5.2.3.7.1
Reference path: resource_with_representation<=
generic_manufacturing_resource<=
action_resource

### 5.1.4.10.1 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

### 5.1.4.11 Tool_assembly

AIM element: tool_assembly
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: tool_assembly<=
action_resource

### 5.1.4.11.1 id

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: tool_assembly<=
action_resource
action_resource.name

### 5.1.4.11.2 number_of_spares

AIM element: count_measure
Source: ISO 10303-41
Reference path: tool_assembly<=
action_resource
action_resource=characterized_resource_definition
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=’number of spares’}
measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure

 

### 5.1.4.11.3 tool_assembly to shape_aspect ( as defined_shape)

AIM element: PATH
Reference path: tool_assembly<=
action_resource <-
requirement_for_action_resource.resources [i]
requirement_for_action_resource <=
action_resource_requirement
action_resource_requirement.operations[i] ->
characterized_action_definition
characterized_action_definition=action
action =>
property_process<-
process_property_association.process
process_property_association
{process_property_association.description='tool assembly shape'}
process_property_association.property_or_shape ->
property_or_shape_select
property_or_shape_select=shape_definition
shape_definition=shape_aspect
shape_aspect

### 5.1.4.11.4 tool_assembly to design_reference (as documented_by)

AIM element: PATH
Reference path: tool_assembly
document_reference_item=tool_assembly
document_reference_item<-
applied_document_reference.items [i]
applied_document_reference<=
document_reference
document_reference.assigned_document->
document =>
design_reference

### 5.1.4.11.5 tool_assembly to tool_assembly_element (as composed_of)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: tool_assembly<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
tool_assembly_relationship
action_resource_relationship.related_resource->
action_resource=>
tool_assembly_element

### 5.1.4.11.6 tool_assembly to tool_ contract (as identified_in_contract)

AIM element: PATH
Reference path: tool_assembly
contract_item=tool_assembly
contract_item <-
applied_contract_assignment.items [i]
applied_contract_assignment <=
contract_assignment
contract_assignment.assigned_contract ->
contract=>
tool_contract

### 5.1.4.11.7 tool_assembly to Tool_placement_instruction (as identifying_data)

AIM element: PATH
Reference path: tool_assembly<=
action_resource
action_resource<-
requirement_for_action_resource.resources[i]
requirement_for_action_resource
action_resource_requirement
characterized_resource_definition=action_resource_requirement
characterized_resource_definition<-
resource_property.resource
resource_property=>
tool_placement_instruction

### 5.1.4.11.8 tool_assembly to property_parameter (as tool_parameters)

AIM element: PATH
Source: ISO 10303-41
Reference path: tool_assembly<=
action_resource
action_resource=characterized_resource_definition
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='tool assembly tool parameteres'}
(descriptive_representation_item)
(measure_representation_item)

 

### 5.1.4.12 Tool_assembly_element

AIM element: tool_assembly_element
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: tool_assembly_element<=
action_resource

### 5.1.4.12.1 catalogue_number

AIM element: document.id
Source: ISO 10303-41
Reference path: tool_assembly_element
document_reference_item=tool_assembly_element
document_reference_item <-
applied_document_reference.items [i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document
{document.name='tool catalogue number'}
document.id

### 5.1.4.12.2 company_name

AIM element: organization.name
Source: ISO 10303-41
Reference path: tool_assembly_element
organization_item=tool_assembly_element
organization_item <-
applied_organization_assignment.items [i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role
organization_role
organization_role.name='tool company name'}
organization_assignment.assigned_organization ->
organization
{organization.description='tool company name'}
organization.name

### 5.1.4.12.3 description

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: tool_assembly_element<=
action_resource
action_resource.description

### 5.1.4.12.4 id

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: tool_assembly_element<=
action_resource
action_resource=characterized_resource_definition
characterized_resource_definition <-
resource_property.resource
resource_property
{resource_property.name='tool element identification'}
resource_property.description

### 5.1.4.12.5 name

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: tool_assembly_element<=
action_resource
action_resource.name

### 5.1.4.12.6 tool_assembly_element to specification (as tool_property)

AIM element: PATH
Reference path: tool_assembly_element
document_reference_item=tool_assembly_element<-
applied_document_reference.items [i]
applied_document_reference<=
document_reference
document_reference.assigned_document->
document=>
process_plan_specification

### 5.1.4.12.7 tool_assembly_element to externally_defined_representation (as tool_representation)

AIM element: PATH
Reference path: tool_assembly_element<=
action_resource
action_resource=characterized_resource_definition
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

 

### 5.1.4.12.8 tool_assembly_element to shape_aspect ( as defined_shape)

AIM element: PATH
Reference path: tool_assembly_element<=
action_resource <-
requirement_for_action_resource.resources [i]
requirement_for_action_resource <=
action_resource_requirement
action_resource_requirement.operations[i] ->
characterized_action_definition=action
action =>
property_process<-
process_property_association.process
process_property_association
{process_property_association.description='tool assembly element shape'}
process_property_association.property_or_shape ->
property_or_shape_select=shape_definition
shape_definition=shape_aspect
shape_aspect

### 5.1.4.13 Tool_body

AIM element: tool_body
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: tool_body
tool_assembly_element<=
action_resource

### 5.1.4.13.1 tool_body to property_parameter (as tool_body_parameter)

AIM element: PATH
Source: ISO 10303-41
Reference path: tool_body<=
tool_assembly_element<=
action_resource
action_resource=characterized_resource_definition
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='tool body parameters'}
(descriptive_representation_item)
(measure_representation_item)

## 5.1.4.14 Tool_magazine_turret_carousel

AIM element: tool_magazine_turret_carousel
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: tool_magazine_turret_carousel<=
action_resource

## 5.1.4.14.1 id

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: tool_magazine_turret_carousel<=
action_resource
action_resource.name

## 5.1.4.14.2 tool_capacity

AIM element: resource_property.description
Source: ISO 10303-49
Reference path: tool_magazine_turret_carousel<=
action_resource
characterized_resource_definition=action_resource
characterized_resource_definition <-
resource_property.resource
resource_property
{resource_property.name='tool_capacity'}
resource_property.description

## 5.1.4.14.3 tool_magazine_turret_carousel to workstation (as used_in)

AIM element: PATH
Rule: 5.2.3.7.2
Reference path: tool_magazine_turret_carousel<=
action_resource <-
action_resource_relationship.relating_resource
action_resource_relationship =>
tool_assembly_relationship
action_resource_relationship.related_resource->
action_resource=>
workstation

 

### 5.1.4.15 Work_cell

AIM element: [work_cell]
[action_resource]
Source: ISO 10303-240
ISO 10303-240
Rule: 5.2.3.7.1
Reference path: [work_cell<=
applied_group_assignment<=
group_assignment]
[action_resource]

### 5.1.4.15.1 description

AIM element: group.description
Source: ISO 10303-41
Reference path: work_cell<=
applied_group_assignment<=
group_assignment
group_assignment.assigned_group->
group
group.description

### 5.1.4.15.2 id

AIM element: group.name
Source: ISO 10303-41
Reference path: work_cell<=
applied_group_assignment<=
group_assignment
group_assignment.assigned_group->
group
group.name

### 5.1.4.15.3 work_cell to workstation (as contains)

AIM element: PATH
Reference path: work_cell<=
applied_group_assignment
{applied_group_assignment<=
group_assignment}
applied_group_assignment.items [i]->
group_item
group_item=workstation
workstation

### 5.1.4.15.4 work_cell to activity (as referenced_by)

AIM element: PATH
Reference path: work_cell<=
action_resource
action_resource.usage[i]->
supported_item
supported_item=action_method
action_method=>
process_plan_activity

### 5.1.4.16 Workstation

AIM element: workstation
Source: ISO 10303-240
Rule: 5.2.3.7.1
Reference path: workstation<=
action_resource

### 5.1.4.16.1 description

AIM element: action_resource.description
Source: ISO 10303-41
Reference path: workstation<=
action_resource
action_resource.description

### 5.1.4.16.2 id

AIM element: action_resource.name
Source: ISO 10303-41
Reference path: workstation<=
action_resource
action_resource.name

## 5.1.5 manufacturing_part_properties UoF

### 5.1.5.1 Alternate_material

AIM element: product_definition
Source: ISO 10303-41
Rules: 5.2.3.5

 

#### 5.1.5.1.1 alternate_ranking

AIM element: make_from_usage_option.ranking
Source: ISO 10303-44
Reference path: product_definition <-
product_definition_relationship.related_product_definition
product_definition_relationship =>
product_definition_usage =>
make_from_usage_option
make_from_usage_option.ranking

#### 5.1.5.1.2 alternate_material to material (as material_substitute)

AIM element: PATH
Reference path: product_definition <-
product_definition_relationship.related_product_definition
{product_definition_relationship =>
product_definition_usage =>
make_from_usage_option}
product_definition_relationship
product_definition_relationship.relating_product_definition ->
product_definition

### 5.1.5.2 Descriptive_parameter

AIM element: descriptive_representation_item
Source: ISO 10303-45

#### 5.1.5.2.1 descriptive_string

AIM element: descriptive_representation_item.description
Source: ISO 10303-45

### 5.1.5.3 Hardness

AIM element: material_property_representation
Source: ISO 10303-45
Reference path: {material_property_representation
material_property_representation.dependent_environment->
data_environment}

### 5.1.5.3.1 high_value

AIM element: measure_representation_item
Source: ISO 10303-45
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

### 5.1.5.3.2 low_value

AIM element: measure_representation_item
Source: ISO 10303-45
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

 

### 5.1.5.3.3 nominal

AIM element: measure_representation_item
Source: ISO 10303-45
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

### 5.1.5.3.4 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

## 5.1.5.4 Material

AIM element: product_definition
Source: ISO 10303-41
Reference path: product_definition
product_definition.name=’material’
{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]}

### 5.1.5.4.1 description

AIM element: product_definition.description
Source: ISO 10303-41

### 5.1.5.4.2 name

AIM element: product_definition.id
Source: ISO 10303-41

### 5.1.5.4.3 order_source

AIM element: organization
Source: ISO 10303-41
Reference path: product_definition
organization_item=product_definition
organization_item <-
applied_organization_assignment.items [i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name='order_source'}
organization_assignment.assigned_organization ->
organization

### 5.1.5.4.4 source_controlled

AIM element: organization
Source: ISO 10303-41
Reference path: product_definition
organization_item=product_definition
organization_item<-
applied_organization_assignment.items [i]
applied_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
organization_role.name='source_controlled'}
organization_assignment.assigned_organization ->
organization

 

### 5.1.5.4.5 material to design_reference (as identified_by_drawing)

AIM element: PATH
Reference path: product_definition
document_reference_item=product_definition
document_reference_item<-
applied_document_reference.items[i]
applied_document_reference<=
document_referenece
document_reference.assigned_document ->
document =>
design_reference

### 5.1.5.4.6 material to base_shape (as initial_material_shape)

AIM element: PATH
Reference path: product_definition
characterized_product_definition=product_definition
characterized_definition=characterized_product_definition<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape

### 5.1.5.4.7 material to shape_description (as initial_material_shape)

AIM element: PATH
Reference path: product_definition
product_definition.formation - >
product_definition_formation
product_definition_formation.description

### 5.1.5.4.8 material to material_property (as material_characteristics)

AIM element: PATH
Reference path: product_definition
characterized_product_definition=product_definition
characterized_definition=characterized_product_definition<-
property_definition.definition
property_definition
property_definition=>
material_property

### 5.1.5.4.9 material to material_specification (as documented_by)

AIM element: PATH
Reference path: product_definition
document_reference_item=product_definition<-
applied_document_reference.items[i]
applied_document_reference<=
document_referenece
document_reference.assigned_document ->
document

## 5.1.5.5 Material_property

AIM element: material_property
Source: ISO 10303-45

### 5.1.5.5.1 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

### 5.1.5.5.2 material_property to property_parameter (as material_parameters)

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)

## 5.1.5.6 Numeric_parameter

AIM element: measure_representation_item
Source: ISO 10303-45

 

#### 5.1.5.6.1 parameter_unit

| | |
|---|---|
| AIM element: | named_unit |
| Source: | ISO 10303-41 |
| Rules: | 5.2.3.4.6 |
| Reference path: | measure_representation_item <= measure_with_unit<br>measure_with_unit.unit_component -> unit<br>unit = named_unit<br>named_unit |

#### 5.1.5.6.2 parameter_value

| | |
|---|---|
| AIM element: | measure_value |
| Source: | ISO 10303-41 |
| Reference path: | measure_representation_item <= measure_with_unit<br>measure_with_unit.value_component -> measure_value |

### 5.1.5.7 Numeric_parameter_with_tolerance

| | |
|---|---|
| AIM element: | [measure_representation_item]<br>[qualified_representation_item] |
| Source: | ISO 10303-45<br>ISO 10303-45 |

#### 5.1.5.7.1 numeric_parameter_with_tolerance to plus_minus_value (as implicit_tolerance)

AIM element: IDENTICAL MAPPING

#### 5.1.5.7.2 numeric_parameter_with_tolerance to tolerance_limit (as implicit_-tolerance)

| | |
|---|---|
| AIM element: | PATH |
| Reference path: | qualified_representation_item<br>qualified_representation_item.qualifiers[i] -> value_qualifier<br>value_qualifier = type_qualifier<br>type_qualifier |

### 5.1.5.8 Part_property

AIM element: property_definition
Source: ISO 10303-41
Reference path: {property_definition
property_definition.name = 'part property'}

#### 5.1.5.8.1 part_property to property_parameter (as part_characteristic)

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)

### 5.1.5.9 Process_property

AIM element: property_definition
Source: ISO 10303-41
Reference path: {property_definition
property_definition.name = 'process property'}

#### 5.1.5.9.1 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

 

#### 5.1.5.9.2 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)

### 5.1.5.10 Property

AIM element: property_definition
Source: ISO 10303-41

#### 5.1.5.10.1 property_name

AIM element: property_definition.name
Source: ISO 10303-41

#### 5.1.5.10.2 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

#### 5.1.5.10.3 property to part_property (as part_property_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'}

### 5.1.5.10.4 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'}

### 5.1.5.10.5 property to shape_aspect (as 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

### 5.1.5.10.6 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

### 5.1.5.10.7 property to surface_property (as surface_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 = 'surface property'}

 

### 5.1.5.11 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

### 5.1.5.11.1 parameter_name

AIM element: representation_item.name
Source: ISO 10303-43

### 5.1.5.12 Surface_property

AIM element: property_definition
Source: ISO 10303-41
Reference path: {property_definition
property_definition.name = 'surface property'}

### 5.1.5.12.1 surface_finish

AIM element: property_definition.description
Source: ISO 10303-41
Reference path: {property_definition
property_definition.description = 'surface finish'}

### 5.1.5.12.2 surface_property to property_parameter (as surface_-characteristics)

AIM element: PATH
Reference path: property_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)

## 5.1.6 manufacturing_process_requirement_documents UoF

### 5.1.6.1 Contract

AIM element: part_contract
Source: ISO 10303-240
Rule: 5.2.3.7.4
Reference path: part_contract<=
contract

#### 5.1.6.1.1 contract_number

AIM element: contract.name
Source: ISO 10303-41
Reference path: part_contract<=
contract
contract.name

#### 5.1.6.1.2 project_number

AIM element: organizational_project.name
Source: ISO 10303-41
Reference path: part_contract
organization_item=part_contract
organization_item<-
applied_organization_assignment.items [i]
applied_organization_assignment<=
organization_assignment
organization_assignment.assigned_organization->
organization <-
organizational_project.responsible_organizations [i]
organizational_project
organizational_project.description=’project number’
organizational_project.name

#### 5.1.6.1.3 contract to part_version (as acquires)

AIM element: PATH
Reference path: part_contract<=
contract<-
contract_assignment.assigned_contract
contract_assignment=>
applied_contract_assignment
applied_contract_assignment.items [i]->
contract_item
contract_item = product_definition_formation
product_definition_formation

 

### 5.1.6.1.4 contract to company_contract_assignment (as binding_agreement)

AIM element: PATH
Reference path: part_contract
organization_item=part_contract
organization_item<-
applied_organization_assignment.items[i]
applied_organization_assignment < =
{organization_assignment
organization_assignment.assigned_organization - >
organization
(organization.description=’binding agreement’)}

### 5.1.6.1.5 contract to document_file (as contract_document)

AIM element: PATH
Reference path: part_contract<=
contract<-
contract_assignment.assigned_contract
contract_assignment=>
applied_contract_assignment
applied_contract_assignment.items [i]->
contract_item
contract_item = document_file
document_file<=
{document
document.description=’contract document’}

## 5.1.6.2 Controller_program

AIM element: controller_program
Source: ISO 10303-240
Rule: 5.2.3.7.5
Reference path: controller_program<=
document

### 5.1.6.2.1 id

AIM element: document.id
Source: ISO 10303-41
Reference path: controller_program<=
document
document.id

### 5.1.6.2.2 controller_type

AIM element: document.name
Source: ISO 10303-41
Reference path: controller_program<=
document
document.name

### 5.1.6.2.3 program_format

AIM element: document.description
Source: ISO 10303-41
Reference path: controller_program<=
document
document.description

### 5.1.6.2.4 revision

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41
Reference path: controller_program<=
document <-
document_usage_constraint.source
document_usage_constraint
{document_usage_constraint.subject_element=
'controller program revision level'}
document_usage_constraint.subject_element_value

### 5.1.6.2.5 validation_date

AIM element: date
Source: ISO 10303-41
Rules: 5.2.3.4.2
Reference path: controller_program<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item=date_and_time
date_and_time
date_and_time.date_component->
date

 

### 5.1.6.2.6 validation_time

AIM element: local_time
Source: ISO 10303-41
Reference path: controller_program<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item=date_and_time
date_and_time
date_and_time.time_component->
local_time

### 5.1.6.2.7 controller_program to manufacturing_activity (as accomplished_by)

AIM element: PATH
Reference path: controller_program<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item
document_reference_item= manufacturing_activity
manufacturing_activity

### 5.1.6.2.8 controller_program to document_file (as controller_data_file)

AIM element: IDENTICAL MAPPING

### 5.1.6.2.9 controller_program to part_version (as identified_on)

AIM element: PATH
Reference path: controller_program<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item
document_reference_item=product_definition_formation
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’ }

### 5.1.6.2.10 controller_program to manufacturing_process (as covers)

AIMelement: PATH
ReferencePath: controller_program<=
document<-
document_reference.assigned_document
document_reference=>
applied_document_reference
applied_document_reference.items[i]->
document_reference_item
document_reference_item=manufacturing_process
manufacturing_process

### 5.1.6.2.11 controller_program to tool_assembly (as references)

AIM element: PATH
Reference path: controller_program<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item
document_reference_item=tool_assembly
tool_assembly

 

### 5.1.6.3 Design_reference

AIM element: design_reference
Source: ISO 10303-240
Rule: 5.2.3.7.5
Reference path: design_reference<=
document

#### 5.1.6.3.1 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

#### 5.1.6.3.2 id

AIM element: document.id
Source: ISO 10303-41
Reference path: design_reference<=
document
document.id

#### 5.1.6.3.3 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

#### 5.1.6.3.4 design_reference to document_file (as drawing_data_file)

AIM element: IDENTICAL MAPPING

### 5.1.6.3.5 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

### 5.1.6.4 Digital_file

AIM element: document_file
Source: ISO 10303-41
Rule: 5.2.3.7.3, 5.2.3.7.5
Reference path: document_file =>
[document<-
{document_representation_type.represented_document
document_representation_type
document_representation_type. name = 'digital'}]
[characterized_object]

### 5.1.6.5 Document_assignment

#1: as document_assignment

AIM element: applied_document_reference
Source: ISO 10303-240
Reference path: applied_document_reference<=
document_reference

#2: as partial_document_assignment

AIM element: applied_document_usage_constraint_assignment
Source: ISO 10303-240
Reference path: applied_document_usage_constraint_assignment <=
document_usage_constraint_assignment

 

### 5.1.6.5.1 role

#1: as document_assignment

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

#2: as partial_document_assignment

AIM element: document_usage_role.name
Source: ISO 10303-41
Reference path: applied_document_usage_constraint_assignment <=
document_usage_constraint_assignment
document_usage_constraint_assignment.role->
document_usage_role
document_usage_role.name

### 5.1.6.5.2 document_assignment to document_file (as assigned_document)

#1: as document_assignment

AIM element: PATH
Reference path: applied_document_reference<=
document_reference
document_reference.assigned_document->
document=>
document_file

#2: as partial_document_assignment

AIM element: PATH
Reference path: applied_document_usage_constraint_assignment<=
document_usage_constraint_assignment
document_usage_constraint_assignment.assigned_document_usage->
document_usage_constraint
document_usage_constraint.source->
document=>
document_file

### 5.1.6.6 Document_file

AIM element: document_file
Source: ISO 10303-41
Rule: 5.2.3.7.3, 5.2.3.7.5
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]

### 5.1.6.6.1 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

### 5.1.6.6.2 file_id

AIM element: document.id
Source: ISO 10303-41
Reference path: document_file<=
document
document.id

### 5.1.6.6.3 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

 

### 5.1.6.6.4 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'}

### 5.1.6.6.5 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}

## 5.1.6.7 Document_file_properties

AIM element: representation
Source: ISO 10303-43

### 5.1.6.7.1 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

### 5.1.6.7.2 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

### 5.1.6.7.3 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

### 5.1.6.7.4 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

### 5.1.6.7.5 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

 

### 5.1.6.7.6 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

### 5.1.6.7.7 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

### 5.1.6.7.8 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

### 5.1.6.7.9 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}

### 5.1.6.7.10 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 caracter code’}
descriptive_representation_item
descriptive_representation_item.description

### 5.1.6.7.11 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}

## 5.1.6.8 Document_file_relationship

AIM element: document_relationship
Source: ISO 10303-41

 

### 5.1.6.8.1 previous_file

AIM element: document_file
Source: ISO 10303-41
Reference path: document_relationship
document_relationship.related_document->
document=>
document_file

### 5.1.6.8.2 succeeding_file

AIM element: document_file
Source: ISO 10303-41
Reference path: document_relationship
document_representation.relating_document->
document=>
document_file

### 5.1.6.9 Executable

AIM element: externally_defined_schema
Source: ISO 10303-41
Reference path: externally_defined_schema<=
externally_defined_item
externally_defined_item.item_id='executable'

### 5.1.6.9.1 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

### 5.1.6.10 External_file_identification

AIM element: applied_external_identification_assignment
Source: ISO 10303-240
Reference path: applied_external_identification_assignment<=
external_identification_assignment<=
identification_assignment

### 5.1.6.10.1 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

### 5.1.6.10.2 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

### 5.1.6.11 External_schema_definition

AIM element: externally_defined_schema
Source: ISO 10303-240
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'}]

### 5.1.6.11.1 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’}

 

### 5.1.6.12 Fixture_contract

AIM element: fixture_contract
Source: ISO 10303-240
Rule: 5.2.3.7.4
Reference path: fixture_contract<=
contract

### 5.1.6.12.1 contract_number

AIM element: contract.name
Source: ISO 10303-41
Reference path: fixture_contract<=
contract
contract.name

### 5.1.6.12.2 design_order

AIM element: versioned_action_request.id
Source: ISO 10303-41
Reference path: fixture_contract
action_request_item=fixture_contract
action_request_item <-
applied_action_request_assignment.items [i]
applied_action_request_assignment <=
action_request_assignment
action_request_assignment.assigned_action_request ->
versioned_action_request
(versioned_action_request.description='design order')
versioned_action_request.id

### 5.1.6.12.3 fabrication_order

AIM element: versioned_action_request.id
Source: ISO 10303-41
Reference path: fixture_contract
action_request_item=fixture_contract
action_request_item <-
applied_action_request_assignment.items [i]
applied_action_request_assignment <=
action_request_assignment
action_request_assignment.assigned_action_request ->
versioned_action_request
(versioned_action_request.description='fabrication order')
versioned_action_request.id

### 5.1.6.12.4 part_number

AIM element: contract_type.description
Source: ISO 10303-41
Reference path: fixture_contract<=
contract
contract.kind->
contract_type
contract_type.description

### 5.1.6.12.5 fixture_contract to in_facility_location (as fixture_location)

AIM element: PATH
Reference path: fixture_contract<=
contract<-
contract_assignment.assigned_contract
contract_assignment=>
applied_contract_assignment
applied_contract_assignment.items [i]->
contract_item
contract_item = in_facility_location
in_facility_location

### 5.1.6.13 Hardcopy

AIM element: document_file
Source: ISO 10303-41
Rule: 5.2.3.7.3, 5.2.3.7.5
Reference path: document_file =>
[document<-
{document_representation_type.represented_documemnt
document_representation_type
document_representation_type. name = 'physical'}]
[characterized_object]

### 5.1.6.14 Illustration

AIM element: illustration
Source: ISO 10303-240
Rule: 5.2.3.7.5
Reference path: illustration<=
document

 

### 5.1.6.14.1 description

AIM element: document.description
Source: ISO 10303-41
Reference path: illustration<=
document
document.description

### 5.1.6.14.2 id

AIM element: document.id
Source: ISO 10303-41
Reference path: illustration<=
document
document.id

### 5.1.6.14.3 illustration to activity (as clarifies)

AIM element: PATH
Reference path: illustration<=
document <-
document_reference.assigned_document
document_reference =>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item= process_plan_activity
process_plan_activity

### 5.1.6.14.4 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

## 5.1.6.15 Material_specification

AIM element: document
Source: ISO 10303-41

### 5.1.6.15.1 procurement_data

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41
Reference path: document<-
document_usage_constraint.source
document_usage_constraint
{document_usage_constraint.subject_element=’ procurement data’}
document_usage_constraint.subject_element_value

## 5.1.6.16 Partial_document_assignment

AIM element: applied_document_usage_constraint_assignment
Source: ISO 10303-240
Reference path: applied_document_usage_constraint_assignment<=
document_usage_constraint_assignment

### 5.1.6.16.1 document_portion

AIM element: document_usage_role.description
Source: ISO 10303-41
Reference path: applied_document_usage_constraint_assignment<=
document_usage_constraint_assignment
document_usage_constraint_assignment.role->
document_usage_role
document_usage_role.description

## 5.1.6.17 Part_dimensioning_standard

AIM element: applied_document_reference
Source: ISO 10303-240
Reference path: applied_document_reference<=
document_reference
{document_reference.role->
object_role
object_role.name= ’dimensioning standard’}

### 5.1.6.17.1 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

 

### 5.1.6.18 Special_capability

AIM element: action_resource_requirement
Source: ISO 10303-41
Reference path: {action_resource_requirement
action_resource_requirement.kind->
resource_requirement_type
resource_requirement_type.description= ‘capability}

### 5.1.6.18.1 name

AIM element: action_resource_requirement.name
Source: ISO 10303-41

### 5.1.6.18.2 specification

AIM element: document
Source: ISO 10303-41
Reference path: action_resource_requirement
document_reference_item=action_resource_requirement
document_reference_item <-
applied_document_reference.items [i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document

### 5.1.6.18.3 special_capability to workstation (as possessed_by)

AIM element:
Source: ISO 10303-41
Reference path: action_resource_requirement =>
requirement_for_action_resource
requirement_for_action_resource.resources [i]->
action_resource=>
workstation

### 5.1.6.19 Special_instruction

AIM element: descriptive_representation_item
Source: ISO 10303-45
Reference path: descriptive_representation_item<=
representation_item
representation_item.name=’special instruction’

### 5.1.6.19.1 instruction_text

AIM element: descriptive_representation_item.description
Source: ISO 10303-41

### 5.1.6.19.2 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

### 5.1.6.20 Specification

AIM element: process_plan_specification
Source: ISO 10303-240
Reference path: process_plan_specification<=
document

### 5.1.6.20.1 description

AIM element: document.description
Source: ISO 10303-41
Reference path: process_plan_specification<=
document
document.description

### 5.1.6.20.2 number_id

AIM element: document.id
Source: ISO 10303-41
Reference path: process_plan_specification<=
document
document.id

### 5.1.6.20.3 revision

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41
Reference path: process_plan_specification<=
document<-
document_usage_constraint.source
document_usage_constraint
{document_usage_constraint.subject_element=’revision’}
document_usage_constraint.subject_element_value

 

### 5.1.6.20.4 subclass

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41
Reference path: process_plan_specification<=
document<-
document_usage_constraint.source
document_usage_constraint
{document_usage_constraint.subject_element='subclass'}
document_usage_constraint.subject_element_value

### 5.1.6.20.5 title

AIM element: document.name
Source: ISO 10303-41
Reference path: process_plan_specification<=
document
document.name

### 5.1.6.20.6 specification to document_assignment (as specification_document)

#1: as document_assingment

AIM element: PATH
Reference path: process_plan_specification<=
document<-
document_reference.assigned_document
document_reference=>
applied_document_reference

#2: as partial_document_assingment

AIM element: PATH
Reference path: process_plan_specification<=
document<-
document_usage_constraint.source
document_usage_constraint<-
document_usage_constraint_assignment.assigned_document_usage
document_usage_constraint_assignment=>
applied_document_usage_constraint_assignment

### 5.1.6.21 Supplemental_document

AIM element: process_plan_specification
Source: ISO 10303-41
Reference path: process_plan_specification<=
document
document.kind->
document_type
document_type.product_data_type='supplemental document'

### 5.1.6.22 Tool_contract

AIM element: tool_contract
Source: ISO 10303-240
Rule: 5.2.3.7.4
Reference path: tool_contract<=
contract

#### 5.1.6.22.1 contract_number

AIM element: contract.name
Source: ISO 10303-41
Reference path: tool_contract<=
contract
contract.name

#### 5.1.6.22.2 design_order

AIM element: versioned_action_request.id
Source: ISO 10303-41
Reference path: tool_contract
action_request_item=tool_contract
action_request_item <-
applied_action_request_assignment.items [i]
applied_action_request_assignment <=
action_request_assignment
action_request_assignment.assigned_action_request ->
versioned_action_request
(versioned_action_request.description='design order')
versioned_action_request.id

 

### 5.1.6.22.3 fabrication_order

AIM element: versioned_action_request.id
Source: ISO 10303-41
Reference path: tool_contract
action_request_item=tool_contract
action_request_item <-
applied_action_request_assignment.items [i]
applied_action_request_assignment <=
action_request_assignment
action_request_assignment.assigned_action_request ->
versioned_action_request
(versioned_action_request.description='fabrication order')
versioned_action_request.id

### 5.1.6.22.4 part_number

AIM element: contract_type.description
Source: ISO 10303-41
Reference path: tool_contract<=
contract
contract.contract_type->
contract_type
contract_type.description

### 5.1.6.22.5 tool_contract to specification (as contract_document)

AIM element: PATH
Reference path: tool_contract<=
contract<-
contract_assignment.assigned_contract
contract_assignment=>
applied_contract_assignment
applied_contract_assignment.items [i]->
contract_item
contract_item = process_plan_specification
process_plan_specification
{process_plan_specification<=
document
document.description='tool contract document'}

### 5.1.6.23 Tool_placement_instruction

AIM element: tool_placement_instruction
Source: ISO 10303-240
Rule: 5.2.3.7.7
Reference path: tool_placement_instruction<=
resource_property

### 5.1.6.23.1 tool_position

AIM element: descriptive_representation_item.description
Source: ISO 10303-41
Reference path: tool_placement_instruction<=
resource_property <-
resource_property_representation.property
resource_property_representation
{resource_property_representation.name='tool_position'}
resource_property_representation.representation ->
representation
representation.items [i]
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

### 5.1.6.23.2 tool_placement_instruction to tool_magazine_turret_carousel ( as configures)

AIM element: PATH
Source: ISO 10303-240
Reference path: tool_placement_instruction<=
resource_property
resource_property.resource->
characterized_resource_definition=action_resource_requirement
action_resource_requirement=>
requirement_for_action_resource
requirement_for_action_resource.resources[i]->
action_resource=>
tool_magazine_turret_carousel

### 5.1.6.23.3 tool_placement_instruction to specification (as instruction_ document)

AIM element: PATH
Source: ISO 10303-41
Reference path: tool_placement_instruction
document_reference_item=tool_placement_instruction
document_reference_item <-
applied_document_reference.items [i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document=>
process_plan_specification

 

### 5.1.6.24 View_reference

AIM element: view_reference
Source: ISO 10303-240
Reference path: view_reference<=
document_usage_constraint

### 5.1.6.24.1 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

### 5.1.6.24.2 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

### 5.1.6.24.3 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

### 5.1.6.24.4 view_reference to activity (as identifies)

AIM element: PATH
Source: ISO 10303-41
Reference path: view_reference<=
document_usage_constraint
document_usage_constraint.source->
document <-
document_reference.assigned_document
document_reference=>
applied_document_reference
applied_document_reference.items [i]->
document_reference_item
document_reference_item=process_plan_activity
process_plan_activity

### 5.1.6.24.5 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

## 5.1.7 measurement_limitations UoF

### 5.1.7.1 Angular_dimension_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
(angular_location)
Source: ISO 10303-224
Rules:
Reference path: (directed_dimensional_location<= )
(angular_location <= )
dimensional_location

#2: if directed = false

AIM element: (angular_location)
Source: ISO 10303-47
Rules: 5.2.3.7

 

#### 5.1.7.1.1 angular_dimension_tolerance to orientation (as plane_and_direction)

#1: if directed = true

AIM element: PATH
Rules: 5.2.3.6
Reference path: (directed_dimensional_location <=
dimensional_location =>
angular_location)

#2: if directed = false

AIM element: PATH
Rule: 5.2.3.6
Reference path: (directed_dimensional_location <=
dimensional_location <=)
shape_aspect_relationship
shape_definition = shape_aspect_relationship
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
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

### 5.1.7.2 Angular_size_dimension_tolerance

AIM element: angular_size
Source: ISO 10303-47

#### 5.1.7.2.1 full_or_half

AIM element: representation_item.name
Source: ISO 10303-43
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
representation_item.name
{(representation_item.name = 'full angle')
(representation_item.name = 'half angle')}

#### 5.1.7.2.2 major_angle

AIM element: angular_size.angle_selection
Source: ISO 10303-47

### 5.1.7.3 Curved_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'curve length'}

### 5.1.7.4 Diameter_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'diameter'}

### 5.1.7.5 Dimensional_tolerance

AIM element: shape_dimension_representation
Source: ISO 10303-47

 

### 5.1.7.5.1 dimension_description

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_size
dimensional_size
dimensional_size.name)
(dimensional_characteristic = dimensional_location
dimensional_location <=
shape_aspect_relationship
shape_aspect_relationship.description)

### 5.1.7.5.2 dimension_note

AIM element: descriptive_representation_item.description
Source: ISO 10303-43
Rules: 5.2.3.6
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= 'auxilary')
(descriptive_representation_item.description= ‘theoretical’)

### 5.1.7.5.3 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

### 5.1.7.5.4 significant_digits

AIM element: precision_qualifier.precision_value
Source: ISO 10303-45
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[i] ->
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

### 5.1.7.5.5 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

### 5.1.7.5.6 dimensional_tolerance to tolerance_value (as limit)

AIM element: PATH
Reference path: shape_dimension_representation <-
dimensional_characteristic_representation.representation
dimensional_characteristic_representation

 

### 5.1.7.6 Distance_along_curve_tolerance

#1: if directed = true

| | |
|---|---|
| AIM element: | (directed_dimensional_location)<br>(dimensional_location_with_path) |
| Source: | ISO 10303-224 |
| Rules: | |
| Reference path: | (dimensional_location_with_path)<br>(directed_dimensional_location <=<br>dimensional_location) |

#2: if directed = false

| | |
|---|---|
| AIM element: | (dimensional_location_with_path) |
| Source: | ISO 10303-47 |
| Rules: | 5.2.3.6 |

#### 5.1.7.6.1 with_curve_direction

| | |
|---|---|
| AIM element: | dimensional_location_with_path.path |
| Source: | ISO 10303-47 |

#### 5.1.7.6.2 distance_along_curve_tolerance to shape_aspect (as path)

| | |
|---|---|
| AIM element: | PATH |
| Reference path: | dimensional_location_with_path<br>dimensional_location_with_path.path -><br>shape_aspect |

## 5.1.7.7 Externally_defined_size_dimension

AIM element: [externally_defined_dimension_definition]
[dimensional_size_with_path]
Source: ISO 10303-240
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 10303-240
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]

### 5.1.7.7.1 tolerance_class

AIM element: dimensional_size.name
Source: ISO 10303-47

 

### 5.1.7.7.2 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'

### 5.1.7.7.3 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

## 5.1.7.8 Geometric_tolerance

AIM element: aic_geometric_tolerances
Source: ISO 10303-519
Rule: 5.2.3.4.13

### 5.1.7.8.1 geometric_tolerance to shape_aspect (as applied_shape)

AIM element: PATH
Reference path: geometric_tolerance
geometric_tolerance.toleranced_shape_aspect ->
shape_aspect

### 5.1.7.9 Height_dimension

#1: if the heigth_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 heigth_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'

#### 5.1.7.9.1 Height_dimension to shape_aspect (as path)

#1: if the heigth_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 heigth_dimension is specified and the optional path is not specified

AIM element: dimensional_size
Source: ISO 10303-47

### 5.1.7.10 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='lenght'

#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='lenght'

 

### 5.1.7.10.1 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

### 5.1.7.11 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

### 5.1.7.11.1 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
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

## 5.1.7.11.2 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
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

## 5.1.7.11.3 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
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

 

## 5.1.7.12 Location_dimension_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
Source: ISO 10303-240
Rule: 5.2.3.6
Reference path: #1: directed_dimensional_location <=
dimensional_location

#2: if directed = false

AIM element: (dimensional_location)
Source: ISO 10303-47
Rule: 5.2.3.6
Reference path: dimensional_location

## 5.1.7.12.1 location_dimension_tolerance to orientation (as plane_and_-direction)

#1: if directed = true

AIM element: PATH
Rule: 5.2.3.6
Reference path: (directed_dimensional_location <=
dimensional_location <=)
shape_aspect_relationship
shape_definition = shape_aspect_relationship
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
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

#2: if directed = false

AIM element: PATH
Rule: 5.2.3.6
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
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 = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

## 5.1.7.13 Location_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
Source: ISO 10303-224
Rules: 5.2.3.6
Reference path: #1: directed_dimensional_location <=
dimensional_location

#2: if directed = false

AIM element: (dimensional_location)
Source: ISO 10303-47
Rules: 5.2.3.6

## 5.1.7.13.1 directed

AIM element: IDENTICAL MAPPING

 

### 5.1.7.13.2 location_tolerance to shape_aspect (as origin_shape)

AIM element: PATH
Reference path: dimensional_location <=
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect

### 5.1.7.13.3 location_tolerance to shape_aspect (as termination_shape)

AIM element: PATH
Reference path: dimensional_location <=
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect

### 5.1.7.14 Machining_tolerance

AIM element: machining_tolerance
Source: ISO 10303-240
Reference path: machining_tolerance<=
action_property

### 5.1.7.14.1 inside_tolerance

AIM element: length_measure_with_unit
Source: ISO 10303-41
Reference path: machining_tolerance<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name='inside_tolerance'}
measure_representation_item <=
measure_with_unit =>
length_measure_with_unit

### 5.1.7.14.2 outside_tolerance

AIM element: length_measure_with_unit
Source: ISO 10303-41
Reference path: machining_tolerance<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name='outside_tolerance'}
measure_representation_item <=
measure_with_unit =>
length_measure_with_unit

### 5.1.7.14.3 total_tolerance

AIM element: length_measure_with_unit
Source: ISO 10303-41
Reference path: machining_tolerance<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name='total_tolerance'}
measure_representation_item <=
measure_with_unit =>
length_measure_with_unit

## 5.1.7.15 Plus_minus_value

#1: if the plus_minus_value is selected for tolerance_value

AIM element: (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

 

### 5.1.7.15.1 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 10303-45
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

### 5.1.7.15.2 significant_digits

#1: if the plus_minus_value is selected for tolerance_value

AIM element: precision_qualifier.precision_value
Source: ISO 10303-45
Reference path: (tolerance_value
[tolerance_value.upper_bound ->]
[tolerance_value.lower_bound ->]
measure_with_unit <-
measure_qualification.qualifiers[i] ->)
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
Reference path: (qualified_representation_item
qualified_representation_item.qualifiers[i] ->)
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

### 5.1.7.15.3 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 10303-45
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

### 5.1.7.16 Radial_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: {dimensional_size
dimensional_size.name = 'radius'}

### 5.1.7.17 Size_tolerance

AIM element: dimensional_size
Source: ISO 10303-47

### 5.1.7.17.1 envelope

AIM element: representation.name
Source: ISO 10303-43
Reference path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <=
shape_representation <=
representation
representation.name
{representation.name = 'envelope tolerance'}

 

### 5.1.7.17.2 size_tolerance to shape_aspect (as applied_shape)

AIM element: PATH
Reference path: dimensional_size
dimensional_size.applies_to ->
shape_aspect

## 5.1.7.18 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'

### 5.1.7.18.1 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

## 5.1.7.19 Tolerance_limit

#1: if the tolerance_limit is selected for tolerance_value

AIM element: type_qualifier
Source: ISO 10303-45

#2: if the tolerance_limit is selected for numeric_parameter

AIM element: qualified_representation_item
Source: ISO 10303-45

## 5.1.7.19.1 limit_qualifier

AIM element: type_qualifier.name
Source: ISO 10303-45
Reference path: (type_qualifier.name='maximum')
(type_qualifier.name='minimum')
(type_qualifier.name)

## 5.1.7.20 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 10303-240
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}

 

### 5.1.7.20.1 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

#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

### 5.1.7.20.2 significant_digits

#1: if the tolerance_range is selected for tolerance_value

AIM element: precision_qualifier.precision_value
Source: ISO 10303-45
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
Reference path: (qualified_representation_item
qualified_representation_item.qualifiers[i] ->)
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

### 5.1.7.20.3 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

### 5.1.7.21 Tolerance_value

AIM element: dimensional_characteristic_representation
Source: ISO 10303-47

 

### 5.1.7.21.1 tolerance_value to limits_and_fits (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 = limits_and_fits
limits_and_fits

### 5.1.7.21.2 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

### 5.1.7.21.3 tolerance_value to tolerance_limit (as defined_value)

AIM element: PATH
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

### 5.1.7.21.4 tolerance_value to tolerance_range (as defined_value)

AIM element: PATH
Reference path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation<=
shape_representation<=
representation
representation.items[i]->
representation_item=>
qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier

### 5.1.7.22 Width_dimension

AIM element: dimensional_size
Source: ISO 10303-47
Reference path: dimensional_size
dimensional_size.name='width dimension'

## 5.1.8 part_model UoF

### 5.1.8.1 Manufactured_assembly

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='assembly definition'}

### 5.1.8.2 Manufactured_assembly_relationship

AIM element: next_assembly_usage_occurrence
Source: ISO 10303-44

 

### 5.1.8.2.1 manufactured_assembly_relationship to manufactured_assembly (as assembly)

AIM element: PATH
Reference path: next_assembly_usage_occurrence<=
assembly_component_usage<=
product_definition_usage<=
product_definition_relationship
product_definition_relationship.relating_product_definition->
product_definition
product_definition.formation->
product_definition_formation

### 5.1.8.2.2 manufactured_assembly_relationship to part_version (as component)

AIM element: PATH
Reference path: next_assembly_usage_occurrence<=
assembly_component_usage<=
product_definition_usage<=
product_definition_relationship
product_definition_relationship.related_product_definition->
product_definition
product_definition.formation->
product_definition_formation

### 5.1.8.2.3 manufactured_assembly_relationship to part_placement (as orientation)

AIM element: PATH
Rule: 5.2.3.5.1
Reference path: next_assembly_usage_occurrence<=
assembly_component_usage<=
product_definition_usage<=
product_definiton_relationship
characterized_product_definition = product_definition_relationship
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
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}
representation.items->
mapped_item

## 5.1.8.3 Mating_definition

AIM element: product_definition
Source: ISO 10303-41
Reference path: product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='mating definition'

### 5.1.8.3.1 mating_type

AIM element: product_definition.name
Source: ISO 10303-41

 

### 5.1.8.3.2 mating_definition to manufactured_assembly (as applied_assembly)

AIM element: PATH
Reference path: product_definition
product_definition.formation->
product_definition_formation
{product_definition_formation<-
product_definition.formation
product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='assembly definition'}

### 5.1.8.3.3 mating_definition to shape_aspect (as mating_ shape)

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 =>
product_definition_shape <-
shape_aspect.of_shape
shape_aspect

### 5.1.8.3.4 mating_definition to single_piece_part (as mating_solution)

AIM element: PATH
Reference path: product_definition<-
product_definition_relationship.relating_product_definition
product_definition_relationship
product_definition_relationship.name='mating solution'
product_definition_relationship.related_product_definition->
product_definition

### 5.1.8.4 Mating_definiton_relationship

AIM element: product_definiton_relationship
Source: ISO 10303-41
Reference path: product_definiton_relationship
product_definiton_relationship.name='mating material'

### 5.1.8.4.1 mating_definition_relationship to mating_definition (as mating_part_definition)

AIM element: PATH
Reference path:
product_definiton_relationship
product_definiton_relationship.relating_product_definition->
product_definition
(product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='mating definition')

### 5.1.8.4.2 mating_definition_relationship to part_placement (as orientation)

AIM element: PATH
Reference path:
product_definiton_relationship
characterized_product_definition = product_definition_relationship
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition
{property_definition =>
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}
representation_map.mapped_representation
representation_map <-
mapped_item.mapping_source->
mapped_item

### 5.1.8.4.3 mating_definition_relationship to single_piece_part (as mated_part)

AIM element: PATH
Reference path:
product_definiton_relationship
product_definiton_relationship.related_product_definition->
product_definition

 

### 5.1.8.5 Mating_relationship

AIM element: product_definiton_relationship
Source: ISO 10303-41
Reference path: product_definiton_relationship
product_definiton_relationship.name='mating membership'

#### 5.1.8.5.1 mating_relationship to single_piece_part (as predecessor)

AIM element: product_definition
Source: ISO 10303-41
Reference path: product_definiton_relationship
product_definiton_relationship.related_product_definition->
product_definition

#### 5.1.8.5.2 mating_relationship to single_piece_part (as successor)

AIM element: product_definition
Source: ISO 10303-41
Reference path: product_definiton_relationship
product_definiton_relationship.relating_product_definition->
product_definition

### 5.1.8.6 Part_version

AIM element: product_definition_formation
Source: ISO 10303-41

#### 5.1.8.6.1 effectivity

AIM element: effectivity
Source: ISO 10303-41
Reference path: product_definition_formation
effectivity_item=product_definition_formation<-
applied_effectivity_assignment.items[i]
applied_effectivity_assignment<=
effectivity_assignment
effectivity_assignment.assigned_effectivity->
effectivity

#### 5.1.8.6.2 nomenclature

AIM element: product.name
Source: ISO 10303-41
Reference path: product_definition_formation
product_definition_formation.of_product ->
product
product.name

### 5.1.8.6.3 part_number

AIM element: product.id
Source: ISO 10303-41
Reference path: product_definition_formation
product_definition_formation.of_product ->
product
product.id

### 5.1.8.6.4 revision_level

AIM element: product_definition_formation.id
Source: ISO 10303-41
Reference path: product_definition_formation
product_definition_formation.id

### 5.1.8.6.5 part_version to alternate_process_plan (as alternate_plan)

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

### 5.1.8.6.6 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

 

### 5.1.8.6.7 part_version to process_plan_version (as fabrication_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

### 5.1.8.6.8 part_version to part_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

### 5.1.8.6.9 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

### 5.1.8.7 Single_piece_part

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'

### 5.1.8.7.1 single_piece_part to material (as material_definition)

AIM element: PATH
Rules: 5.2.3.5
Reference path: product_definition_formation <-
product_definition.formation
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}

 

#### 5.1.8.7.2 single_piece_part to alternate_material (as alternate_material_-definition)

AIM element: PATH
Rules: 5.2.3.5
Reference path: product_definition_formation <-
product_definition.formation
product_definition <-
product_definition_relationship.relating_product_definition
{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]
[product_definition <-
product_definition_relationship.related_product_definition
{product_definition_relationship =>
product_definition_usage =>
make_from_usage_option}
product_definition_relationship
product_definition_relationship.relating_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]}

## 5.1.9 process_activities UoF

### 5.1.9.1 Activity

AIM element: process_plan_activity
Source: ISO 10303-240
Reference path: process_plan_activity<=
action_method

### 5.1.9.1.1 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

### 5.1.9.1.2 description

AIM element: action_method.description
Source: ISO 10303-41
Reference path: process_plan_activity<=
action_method
action_method.description

### 5.1.9.1.3 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

 

### 5.1.9.1.4 name

AIM element: action_method.name
Source: ISO 10303-41
Reference path: process_plan_activity<=
action_method
action_method.name

### 5.1.9.1.5 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

### 5.1.9.1.6 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

### 5.1.9.1.7 activity to supplemental_document (as constrained_by)

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'

### 5.1.9.1.8 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)

### 5.1.9.1.9 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

### 5.1.9.1.10 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

 

### 5.1.9.1.11 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
document <-
document_usage_constraint.source
document_usage_constraint=>
view_reference

### 5.1.9.1.12 activity to material (as references)

AIM element: PATH
Reference path: process_plan_activity<=
action_method <-
action.chosen_method
action =>
product_definition_process <-
process_product_association.process
process_product_association
{process_product_association.description='process plan activity material'}
process_product_association.defined_product->
characterized_product_definition
characterized_product_definition=product_definition
product_definition

### 5.1.9.1.13 activity to generic_manufacturing_resource (as resources)

AIM element: PATH
Reference path: process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage
action_resource=>
generic_manufacturing_resource

## 5.1.9.2 Allowed_time

AIM element: allowed_time
Source: ISO 10303-240
Reference path: allowed_time <=
action_property

### 5.1.9.2.1 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_measuret

### 5.1.9.2.2 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

### 5.1.9.3 Alternate_activity

AIM element: process_plan_activity
Source: ISO 10303-240
Reference path: process_plan_activity<=
action_method

 

### 5.1.9.3.1 primary_activity

AIM element: process_plan_activity
Source: ISO 10303-240
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.relating_method ->
action_method=>
process_plan_activity

### 5.1.9.4 Ancillary_activity

AIM element: ancillary_activity
Source: ISO 10303-240
Reference path: ancillary_activity <=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.4.1 method

AIM element: action_method.purpose
Source: ISO 10303-41
Reference path: ancillary_activity <=
manufacturing_activity<=
process_plan_activity<=
action_method
action_method.purpose

### 5.1.9.5 Ancillary_setup

AIM element: ancillary_setup
Source: ISO 10303-240
Reference path: ancillary_setup <=
process_plan_activity<=
action_method

### 5.1.9.5.1 setup_type

AIM element: action_method.purpose
Source: ISO 10303-41
Reference path: ancillary_setup <=
process_plan_activity<=
action_method
action_method.purpose

### 5.1.9.6 Continuous_process

AIM element: continuous_process_relationship
Source: ISO 10303-240
Reference path: continuous_process_relationship<=
sequential_method<=
serial_action_method<=
action_method_relationship

### 5.1.9.6.1 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‘)}

### 5.1.9.6.2 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

### 5.1.9.6.3 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.relating_method->
action_method=>
manufacturing_process

 

### 5.1.9.7 Fixture_machine_mounting

AIM element: fixture_machine_mounting
Source: ISO 10303-240
Reference path: fixture_machine_mounting <=
fixture_setup <=
process_plan_activity<=
action_method

### 5.1.9.8 Fixture_machine_unmounting

AIM element: fixture_machine_unmounting
Source: ISO 10303-240
Reference path: fixture_machine_unmounting <=
fixture_setup <=
process_plan_activity<=
action_method

### 5.1.9.9 Fixture_pallet_mounting

AIM element: fixture_pallet_mounting
Source: ISO 10303-240
Reference path: fixture_pallet_mounting <=
fixture_setup <=
process_plan_activity<=
action_method

### 5.1.9.10 Fixture_pallet_unmounting

AIM element: fixture_pallet_unmounting
Source: ISO 10303-240
Reference path: fixture_pallet_unmounting <=
fixture_setup <=
process_plan_activity<=
action_method

### 5.1.9.11 Fixture_setup

AIM element: fixture_setup
Source: ISO 10303-240
Reference path: fixture_setup <=
process_plan_activity<=
action_method

### 5.1.9.11.1 fixture_setup to mounting_position (as is_positioned_on)

AIM element: PATH
Reference path: fixture_setup <=
process_plan_activity<=
action_method
action_method=characterized_action_definition
characterized_action_definition <-
action_property.definition
action_property=>
mounting_position

### 5.1.9.11.2 fixture_setup to machine (as identifies)

AIM element: PATH
Reference path: fixture_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

### 5.1.9.11.3 fixture_setup to fixture_assembly (as identifies_fixture)

AIM element: PATH
Reference path: fixture_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=>
fixture_assembly

 

### 5.1.9.11.4 fixture_setup to pallet (as identifies_pallet)

AIM element: PATH
Reference path: fixture_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=>
pallet

## 5.1.9.12 Machine_parameters

AIM element: representation_item
Source: ISO 10303-43

### 5.1.9.12.1 table_indexing_function

AIM element: descriptive_representation_item
Source: ISO 10303-45
Reference path: representation_item =>
{representation_item.name=' table indexing function'}
descriptive_representation_item
{(descriptive_representation_item.description = ‘TRUE‘)
(descriptive_representation_item.description = ‘FALSE‘)
(descriptive_representation_item.description = ‘UNKNOWN‘)}

### 5.1.9.12.2 machine_parameters to numeric_parameter (as maximum_spindle_-speed)

AIM element: PATH
Reference path: representation_item
{representation_item.name=' maximum spindle speed'
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
ratio_measure_with_unit}

### 5.1.9.12.3 machine_parameters to numeric_parameter (as maximum_feedrate)

AIM element: PATH
Reference path: representation_item
{representation_item.name='maximum feedrate'
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}}

### 5.1.9.12.4 machine_parameters to numeric_parameter (as number_of_-control_axis)

AIM element: PATH
Reference path: representation_item
{representation_item.name='number of control axis'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
count_measure}

### 5.1.9.12.5 machine_parameters to numeric_parameter (as table_size)

AIM element: PATH
Reference path: representation_item
{(representation_item.name='x-axis table size')
(representation_item.name='y-axis table size')}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

### 5.1.9.12.6 machine_parameters to numeric_parameter (as position_accuracy)

AIM element: PATH
Reference path: representation_item
{representation_item.name='position accuracy'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

 

### 5.1.9.12.7 machine_parameters to numeric_parameter (as axis_range_of_motion)

AIM element: PATH
Reference path: representation_item
{representation_item.name='axis range of motion'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

### 5.1.9.12.8 machine_parameters to discriptive_parameter (as axis_range_of_-motion_description)

AIM element: PATH
Reference path: representation_item
{representation_item.name='axis range of motion description'}
representation_item =>
descriptive_representation_item

### 5.1.9.12.9 machine_parameters to numeric_parameter (as number_-of_simultaneous_axis)

AIM element: PATH
Reference path: representation_item
{representation_item.name='number of simultaneous axis'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>>
count_measure}

### 5.1.9.12.10 machine_parameters to numeric_parameter (as spindle_power)

AIM element: PATH
Reference path: representation_item
{representation_item.name='spindle power'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
ratio_measure_with_unit}

### 5.1.9.12.11 machine_parameters to property_parameter (as machine_-parameters)

AIM element: IDENTICAL MAPPING

### 5.1.9.13 Machine_setup

AIM element: machine_setup
Source: ISO 10303-240
Reference path: machine_setup <=
process_plan_activity<=
action_method

### 5.1.9.13.1 setup_instruction

AIM element: action_method.description
Source: ISO 10303-41
Reference path: machine_setup <=
process_plan_activity<=
action_method
action_method.description

### 5.1.9.13.2 machine_setup to machine (as identifies)

AIM element: PATH
Reference path: 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

### 5.1.9.14 Machine_usage

AIM element: machine_usage
Source: ISO 10303-240
Reference path: machine_usage<=
requirement_for_action_resource<=
action_resourec_requirement

### 5.1.9.14.1 machine_class

AIM element: action_resource_requirement.name
Source: ISO 10303-49
Reference path: machine_usage<=
requirement_for_action_resource <=
action_resource_requirement
action_resource_requirement.name

 

### 5.1.9.14.2 machine_usage to machine (as classifies)

AIM element: PATH
Reference path: machine_usage<=
requirement_for_action_resource
requirement_for_action_resources.resources [i]->
action_resource=>
machine

### 5.1.9.14.3 machine_usage to specification (as machine_specification)

AIM element: PATH
Reference path: machine_usage
document_reference_item=machine_usage
document_reference_item <-
applied_document_reference.items [i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document=>
process_plan_specification

### 5.1.9.14.4 machine_usage to machine_parameters (as machine_parameters)

AIM element: PATH
Reference path: machine_usage<=
requirement_for_action_resource<=
action_resource_requirement
action_resource_requirement=characterized_resource_definition
characterized_resource_definition <-
resource_property.resource
resource_property <-
resource_property_representation.property
resource_property_representation
resource_property_representation.representation ->
representation
representation.name=’machine usage parameter’
representation.items [i]->
representation_item

## 5.1.9.15 Machining_process

AIM element: machining_process
Source: ISO 10303-240
Reference path: machining_process<=
manufacturing_process <=
action_method

### 5.1.9.15.1 machining_process to machine (as required_machine)

AIM element: PATH
Reference path: machining_process<=
manufacturing_process<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
machine

## 5.1.9.16 Manufacturing_activity

AIM element: manufacturing_activity
Source: ISO 10303-240
Reference path: manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.16.1 manufacturing_activity to external_schema_definition (as micro_-plan_reference)

AIM element: PATH
Reference path: 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_schema
externally_defined_schema

## 5.1.9.17 Mounting_position

AIM element: mounting_position
Source: ISO 10303-240
Reference path: mounting_position<=
action_property

 

### 5.1.9.17.1 location_origin

AIM element: cartesian_point
Reference path: mounting_position<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name =’location origin’}
geometric_representation_item =>
point =>
cartesian_point

### 5.1.9.17.2 orientation

AIM element: placement
Reference path: mounting_position<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name='orientation'}
geometric_representation_item <=
placement

### 5.1.9.17.3 reference_plane

AIM element: PATH
Reference path: mounting_position<=
action_property <-
action_property_representation.property
action_property_representation
action_property_representation.representation ->
representation
representation.items [i]->
representation_item =>
{representation_item.name='reference plane'}
geometric_representation_item =>
elementary_surface =>
plane

### 5.1.9.18 Non_machining_process

AIM element: non_machining_process<=
manufacturing_process<=
action_method
Source: ISO 10303-240

### 5.1.9.19 Pallet_machine_mounting

AIM element: pallet_machine_mounting
Source: ISO 10303-240
Reference path: pallet_machine_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method

#### 5.1.9.19.1 pallet_machine_mounting to machine (as identifies_machine)

AIM element: PATH
Reference path: pallet_machine_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
machine

#### 5.1.9.19.2 pallet_machine_mounting to pallet (as identifies_pallet)

AIM element: PATH
Reference path: pallet_machine_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
pallet

 

### 5.1.9.20 Pallet_machine_unmounting

AIM element: part_unmounting
Source: ISO 10303-240
Reference path: part_unmounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
(action_method.name=' pallet machine unmounting')

### 5.1.9.21 Part_fixture_mounting

AIM element: part_fixture_mounting
Source: ISO 10303-240
Reference path: part_fixture_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.21.1 part_fixture_mounting to fixture_assembly(as identifies_fixture)

AIM element: PATH
Reference path: part_fixture_mounting<=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
fixture_assembly

### 5.1.9.22 Part_fixture_unmounting

AIM element: part_unmounting
Source: ISO 10303-240
Reference path: part_unmounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
(action_method.name='part fixture unmounting')

## 5.1.9.23 Part_machine_mounting

AIM element: part_machine_mounting
Source: ISO 10303-240
Reference path: part_machine_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.23.1 part_machine_mounting to machine (as identifies_machine)

AIM element: PATH
Reference path: part_machine_mounting <=
part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
machine

## 5.1.9.24 Part_machine_unmounting

AIM element: part_unmounting
Source: ISO 10303-240
Reference path: part_unmounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
(action_method.name='part machine unmounting')

## 5.1.9.25 Part_mounting

AIM element: part_mounting
Source: ISO 10303-240
Reference path: part_mounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
(pallet_machine_mounting)
(part_machne_mounting)
(part_fixture_mounting)

 

### 5.1.9.25.1 part_mounting to part_shape (as is_located_on)

AIM element: PATH
Reference path: part_mounting<=
manufacturing_activity<=
process_plan_activity<=
action_method <-
action.chosen_method
action =>
property_process <-
process_property_association.process
process_property_association
{process_property_association.description='part mounting part shape'}
process_property_association.property_or_shape ->
property_or_shape_select=property_definition
property_definition =>
product_definition_shape

### 5.1.9.25.2 part_mounting to mounting_position (as is_positioned_on)

AIM element: PATH
Reference path: part_mounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
action_method=characterized_action_definition
characterized_action_definition <-
action_property.definition
action_property=>
mounting_position

### 5.1.9.25.3 part_location_origin

AIM element: cartesian_point
Source: ISO 10303-42
Reference path: part_mounting <=
manufacturing_activity<=
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='part location origin'}
geometric_representation_item =>
point =>
cartesian_point

## 5.1.9.26 Part_routing

AIM element: part_routing
Source: ISO 10303-240
Reference path: part_routing <=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.26.1 method

AIM element: action_method.purpose
Source: ISO 10303-41
Reference path: part_routing <=
manufacturing_activity<=
process_plan_activity<=
action_method
action_method.purpose

 

### 5.1.9.27 Part_unmounting

AIM element: part_unmounting
Source: ISO 10303-240
Reference path: part_unmounting <=
manufacturing_activity<=
process_plan_activity<=
action_method
(action_method.name='pallet machine unmounting')
(action_method.name='part fixture unmounting')
(action_method.name='part machine unmounting')

### 5.1.9.28 Performance_rate

AIM element: (allowed_time)
(production_rate)
Source: ISO 10303-240
ISO 10303-240
Reference path: (allowed_time <=)
(production_rate <=)
action_property

### 5.1.9.28.1 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

### 5.1.9.28.2 source

AIM element: action_property.description
Source: ISO 10303-41
Reference path: (allowed_time <=)
(production_rate <=)
action_property
action_property.description

## 5.1.9.29 Process_activity

AIM element: process_activity
Source: ISO 10303-240
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.29.1 type_of_operation

AIM element: action_method.name
Source: ISO 10303-41
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method
action_method.name

### 5.1.9.29.2 process_activity to shape_aspect (as described_by)

AIM element: PATH
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method <-
action.chosen_method
action =>
property_process <-
process_property_association.process
process_property_association
{process_property_association.description='process activity part shape'}
process_property_association.property_or_shape ->
property_or_shape_select
property_or_shape_select = shape_definition
shape_definition =shape_aspect
shape_aspect

 

### 5.1.9.29.3 process_activity to machine_setup (as identifies)

AIM element: PATH
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method<-
action_method_relationship.relating_method
action_method_relationship=>
defining_action_method_relationship
action_method_relationship
action_method_relationship.related_method ->
action_method=>
machine_setup

### 5.1.9.29.4 process_activity to fixture_assembly (as identifies_fixture)

AIM element: PATH
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
fixture_assembly

### 5.1.9.29.5 process_activity to process_property (as process_parameters)

AIM element: PATH
Reference path: process_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_product_definition
characterized_product_definition
characterized_product_definition = product_definition
product_definition
product_definition.formation->
product_definition_formation}

### 5.1.9.29.6 process_activity to machining_tolerance (as tolerances)

AIM element: PATH
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method
characterized_action_definition=action_method
characterized_action_definition<-
action_property.definition
action_property=>
machining_tolerance

### 5.1.9.29.7 process_activity to tool_assembly (as uses_to_perform)

AIM element: PATH
Reference path: process_activity<=
manufacturing_activity<=
process_plan_activity<=
action_method
supported_item=action_method
supported_item <-
action_resource.usage[i]
action_resource =>
tool_assembly

### 5.1.9.30 Production_rate

AIM element: production_rate
Source: ISO 10303-240
Reference path: production_rate <=
action_property

### 5.1.9.30.1 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

 

### 5.1.9.30.2 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

### 5.1.9.31 Setup_activity

AIM element: (ancillary_setup)
(machine_setup)
(fixture_setup)
(tool_setup)
Source: ISO 10303-240
ISO 10303-240
ISO 10303-240
ISO 10303-240
Reference path: (ancillary_setup <=)
(machine_setup<=)
(fixture_setup<=)
(tool_setup<=)
process_plan_activity<=
action_method

### 5.1.9.32 Manufacturing_process

AIM element: manufacturing_process
Source: ISO 10303-240
Reference path: manufacturing_process<=
action_method

### 5.1.9.32.1 description

AIM element: action_method.description
Source: ISO 10303-41
Reference path: manufacturing_process<=
action_method
action_method.description

### 5.1.9.32.2 name

AIM element: action_method.name
Source: ISO 10303-41
Reference path: manufacturing_process<=
action_method
action_method.name

### 5.1.9.32.3 operation_type

AIM element: action_method.purpose
Source: ISO 10303-240
Reference path: manufacturing_process<=
action_method
action_method.purpose
(action_method.purpose='legacy nc')
(action_method.purpose='non nc')
(action_method.purpose='integrated nc')

### 5.1.9.32.4 manufacturing_process to manufacturing_process_feature (as assigned_feature)

AIM element: PATH
Reference path: manufacturing_process<=
action_method<-
action.chosen_method
action =>
property_process<-
process_property_association.process
process_property_association
{process_property_association.name='assigned feature'}
process_property_association.property_or_shape ->
property_or_shape_select
property_or_shape_select = shape_definition
shape_definition =shape_aspect_relationship
shape_aspect_relationship
(shape_aspect_relationship.related_shape_aspect->
shape_aspect)
(shape_aspect_relationship.relating_shape_aspect->
shape_aspect)

 

### 5.1.9.32.5 manufacturing_process to manufacturing_activity (as assigned_operation)

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=>
manufacturing_activity_relationship)
action_method_relationship.related_method->
action_method=>
process_plan_activity=>
manufacturing_activity

### 5.1.9.32.6 manufacturing_process to intermediate_shape (as in_process_shape)

AIM element: PATH
Reference path: manufacturing_process<=
action_method<-
action.chosen_method
action =>
property_process<-
process_property_association.process
process_property_association
process_property_association.description='intermediate shape'
process_property_association.property_or_shape ->
property_or_shape_select
property_or_shape_select = shape_definition
shape_definition =shape_aspect_relationship
shape_aspect_relationship

### 5.1.9.32.7 manufacturing_process to part_holding_position (as part_holding_downs)

AIM element: PATH
Reference path: manufacturing_process<=
action_method
characterized_action_definition=action_method
characterized_action_definition<-
action_resource_requirement.operations
action_resource_requirement
characterized_resource_definition=action_resource_requirement
characterized_resource_definition<-
resource_property.resource
resource_property=>
part_holding_position

### 5.1.9.32.8 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)
(fixture_setup)
(tool_setup)

## 5.1.9.33 Tool_setup

AIM element: tool_setup
Source: ISO 10303-240
Reference path: tool_setup <=
process_plan_activity<=
action_method

### 5.1.9.33.1 tool_setup to machine (as identifies_machine)

AIM element: PATH
Reference path: tool_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

 

### 5.1.9.33.2 tool_setup to tool_assembly (as identifies_tool)

AIM element: PATH
Reference path: tool_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=>
tool_assembly

### 5.1.9.33.3 tool_setup to Tool_magazine_turret_carousel (as tool_placement)

AIM element: PATH
Reference path: tool_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=>
tool_magazine_turret_carousel

### 5.1.9.34 Validation

AIM element: validation
Source: ISO 10303-240
Reference path: validation<=
manufacturing_activity<=
process_plan_activity<=
action_method

### 5.1.9.34.1 method_of_validation

AIM element: action_method.purpose
Source: ISO 10303-49
Reference path: validation<=
manufacturing_activity<=
process_plan_activity<=
action_method
action_method.purpose

### 5.1.9.34.2 validation to controller_program (as checks)

AIM element: PATH
Reference path: validation<=
manufacturing_activity<=
process_plan_activity<=
action_method=>
action_method_with_associated_documents
action_method_with_associated_documents.documents[i]->
document=>
controller_program

### 5.1.9.34.3 validation to fixture_assembly (as identifies)

AIM element: PATH
Reference path: validation<=
manufacturing_activity<=
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=>
fixture_assembly

### 5.1.9.34.4 validation to tool_assembly (as uses_to_perform)

AIM element: PATH
Reference path: validation<=
manufacturing_activity<=
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=>
tool_assembly

 

### 5.1.9.34.5 validation to part_shape (as uses_for_validation)

AIM element: PATH
Reference path: validation<=
manufacturing_activity<=
process_plan_activity<=
action_method <-
action.chosen_method
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 =>
product_definition_shape

## 5.1.10 process_plan UoF

### 5.1.10.1 Alternate_process_plan

AIM element: process_plan_version
Source: ISO 10303-240
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')

#### 5.1.10.1.1 alternate_ranking

AIM element: action.name
Source: ISO 10303-41
Reference path: process_plan_version<=
product_definition_process<=
action
action.name

### 5.1.10.1.2 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.relating_action->
action=>
product_definition_process=>
process_plan_version

## 5.1.10.2 Feature_dependency

AIM element: feature_dependency
Source: ISO 10303-240
Rule: 5.2.3.7.6
Reference path: feature_dependency<=
group_relationship

### 5.1.10.2.1 description

AIM element: group_relationship.description
Source: ISO 10303-41
Reference path: feature_dependency<=
group_relationship
group_relationship.description

### 5.1.10.2.2 name

AIM element: group_relationship.name
Source: ISO 10303-41
Reference path: feature_dependency<=
group_relationship
group_relationship.name

### 5.1.10.2.3 feature_dependency to feature_identification_item (as relating_-feature_item)

AIM element: PATH
Reference path: feature_dependency<=
group_relationship
group_relationship.relating_group->
group=>
class=>
feature_identification_item

 

#### 5.1.10.2.4 feature_dependency to feature_identification_item (as related_-feature_item)

AIM element: PATH
Reference path: feature_dependency<=
group_relationship
group_relationship.related_group->
group=>
class=>
feature_identification_item

### 5.1.10.3 Feature_identification_item

AIM element: feature_identification_item
Source: ISO 10303-240
Reference path: feature_identification_item<=
class<=
group

#### 5.1.10.3.1 feature_identification_item to feature_process (as process)

AIM element: PATH
Reference path: feature_identification_item<=
class<=
group<-
group_assignment.assigned_group
group_assignment=>
applied_group_assignment
applied_group_assignment.items[i]->
group_item=feature_process
feature_process

#### 5.1.10.3.2 feature_identification_item to manufacturing_process_feature (as feature)

AIM element: PATH
Reference path: feature_identification_item<=
class<=
group<-
group_assignment.assigned_group
group_assignment=>
applied_group_assignment
applied_group_assignment.items[i]->
group_item=instanced_feature
instanced_feature

#### 5.1.10.3.3 feature_identification_item to feature_interaction (as feature_-relation)

AIM element: PATH
Reference path: feature_identification_item<=
class<=
group<-
group_relationship.relating_group
group_relationship=>
feature_interaction

### 5.1.10.4 Feature_interaction

AIM element: feature_interaction
Source: ISO 10303-240
Reference path: feature_interaction<=
group_relationship

#### 5.1.10.4.1 description

AIM element: group_relationship.description
Source: ISO 10303-41
Reference path: feature_interaction<=
group_relationship
group_relationship.description

#### 5.1.10.4.2 feature_interaction to feature_identification_item (as related_-feature)

AIM element: PATH
Reference path: feature_interaction<=
group_relationship
group_relationship.related_group->
group=>
class=>
feature_identification_item

### 5.1.10.5 Feature_process

AIM element: feature_process
Source: ISO 10303-240
Reference path: feature_process<=
action_method

 

### 5.1.10.5.1 feature_process to manufacturing_activity (as assigned_operation)

AIM element: PATH
Reference path: feature_process<=
action_method
action_method<-
action_method_relationship.relating_method
action_method_relationship
(action_method_relationship=>
serial_action_method=>
sequential_method)
action_method_relationship.related_method->
action_method=>
process_plan_activity=>
manufacturing_activity

### 5.1.10.5.2 feature_process to property_parameter (as parameter_for_process)

AIM element: PATH
Reference path: feature_process<=
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 parameter'}
(descriptive_representation_item)
(measure_representation_item)

### 5.1.10.5.3 feature_process to manufacturing_process (as assigned_process)

AIM element: PATH
Reference path: feature_process<=
action_method
action_method<-
action_method_relationship.relating_method
action_method_relationship
(action_method_relationship=>
serial_action_method)
action_method_relationship.related_method->
action_method=>
manufacturing_process

### 5.1.10.6 Process_plan_security

AIM element: process_plan_security
Source: ISO 10303-240
Reference path: process_plan_security=>
security_classification

### 5.1.10.6.1 classification_date

AIM element: calendar_date
Source: ISO 10303-41
Rules: 5.2.3.4.2
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

### 5.1.10.6.2 declassification_date

AIM element: calendar_date
Source: ISO 10303-41
Rules: 5.2.3.4.2
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

 

### 5.1.10.6.3 security_code

AIM element: security_classification_level.name
Source: ISO 10303-41
Reference path: process_plan_security=>
security_classification
security_classification.security_level ->
security_classification_level
security_classification_level.name

### 5.1.10.6.4 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

### 5.1.10.6.5 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

### 5.1.10.7 Process_plan_version

AIM element: process_plan_version
Source: ISO 10303-240
Reference path: process_plan_version <=
product_definition_process<=
action

### 5.1.10.7.1 description

AIM element: action.description
Source: ISO 10303-41
Reference path: process_plan_version <=
product_definition_process<=
action
action.description

### 5.1.10.7.2 id

AIM element: action.id
Source: ISO 10303-41
Reference path: process_plan_version <=
product_definition_process<=
action
action.id

### 5.1.10.7.3 manufacturing_GT_code

AIM element: action_property.name
Source: ISO 10303-41
Reference path: process_plan_version <=
product_definition_process<=
action
characterized_action_definition = action
characterized_action_definition <-
action_property.definition
action_property
{action_property.description='manufacturing gt code'}
action_property.name

 

### 5.1.10.7.4 quantity_of_parts

AIM element: count_measure
Source: ISO 10303-41
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 = 'quantity of parts'
representation_item =>
measure_representation_item
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure

### 5.1.10.7.5 process_plan_version to feature_dependency (as feature_-dependency_suggestion)

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=feature_dependency
feature_dependency

### 5.1.10.7.6 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.relating_method
action_method_relationship
(action_method_relationship=>
serial_action_method=>
sequential_method=>
manufacturing_process_relationship)
action_method_relationship.related_method->
action_method=>
manufacturing_process}

### 5.1.10.7.7 process_plan_version to property_parameter (as auxiliary_header_-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.item[i]->
representation_item
{representation_item.name='auxiliary header'}
(descriptive_representation_item)
(measure_representation_item)

 

### 5.1.10.7.8 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

### 5.1.10.7.9 process_plan_version to range_of_parts (as quantity_range)

AIM element: PATH
Reference path: process_plan_version <=
product_definition_process <=
action
characterized_action_definition=action
characterized_action_definition<-
action_property.definition
action_property=>
range_of_parts

### 5.1.10.7.10 process_plan_version to material (as required_material)

AIM element: PATH
Reference path: process_plan_version <=
product_definition_process<-
process_product_association.process
process_product_association
(process_product_association.description='base shape')
process_product_association.defined_product ->
characterzed_product_definiton=product_definition
product_definition

### 5.1.10.8 Range_of_parts

AIM element: range_of_parts
Source: ISO 10303-240
Reference path: range_of_parts<=
action_property

### 5.1.10.8.1 low_value

AIM element: count_measure
Source: ISO 10303-41
Reference path: range_of_parts<=
action_property<-
action_property_representation.property->
action_property_representation
action_property_representation.representation->
representation
representation.items[i]->
representation_item=>
{representation_item.name='low value'}
measure_representation_item<=
measure_with_unit
measure_with_unit.value_component->
measure_value
measure_value = count_measure
count_measure

### 5.1.10.8.2 high_value

AIM element: count_measure
Source: ISO 10303-41
Reference path: range_of_parts<=
action_property<-
action_property_representation.property->
action_property_representation
action_property_representation.representation->
representation
representation.items[i]->
representation_item=>
{representation_item.name='high value'}
measure_representation_item<=
measure_with_unit
measure_with_unit.value_component->
measure_value
measure_value = count_measure
count_measure

 

### 5.1.10.9 Revision

AIM element: revision
Source: ISO 10303-240
Reference path: revision<=
action_relationship

### 5.1.10.9.1 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

### 5.1.10.9.2 description

AIM element: action_relationship.description
Source: ISO 10303-41
Reference path: revision<=
action_relationship
action_relationship. description

### 5.1.10.9.3 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

### 5.1.10.9.4 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

### 5.1.10.9.5 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

### 5.1.10.9.6 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

 

## 5.1.11 shape_representation_for_process_planning UoF

### 5.1.11.1 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)}

### 5.1.11.2 Block_base_shape

AIM element: block_shape_representation
Source: ISO 10303-240
Reference path: block_shape_representation <=
shape_representation_with_parameters <=
shape_representation

#### 5.1.11.2.1 block_base_shape to numeric_parameter (as height)

AIM element: PATH
Reference path: block_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
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}

### 5.1.11.2.2 block_base_shape to numeric_parameter (as width)

AIM element: PATH
Reference path: block_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
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}

## 5.1.11.3 Cylindrical_base_shape

AIM element: cylindrical_shape_representation
Source: ISO 10303-240
Rule: 5.2.3.7.9
Reference path: cylindrical_shape_representation <=
shape_representation_with_parameters <=
shape_representation

### 5.1.11.3.1 cylindrical_base_shape to numeric_parameter (as diameter)

AIM element: PATH
Reference path: cylindrical_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
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}

## 5.1.11.4 Explicit_base_shape_representation

AIM element: shape_representation
Source: ISO 10303-41

 

### 5.1.11.4.1 explicit_shape

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: shape_representation<=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'explicit shape'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'casting')
(descriptive_representation_item.description = 'forging')
(descriptive_representation_item.description = 'composite shape')
(descriptive_representation_item.description = 'user defined')}

### 5.1.11.4.2 user_defined_description

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Reference path: shape_representation<=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'user defined description'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

### 5.1.11.4.3 explicit_base_shape_representation to shape_aspect_representation (as shape_form)

AIM element: IDENTICAL MAPPING
Source: ISO 10303-41

### 5.1.11.5 Geometric_model

#1: advanced_boundary_rep

AIM element: [manifold_solid_brep]
Source: ISO 10303-42

#2: faceted_b_rep

AIM element: [faceted_brep]
Source: ISO 10303-42

#3: manifold_surface_with_topology

AIM element: [shell_based_surface_model]
Source: ISO 10303-42

#4:non_topological_surface_and_wireframe

AIM element: [geometric_curve_set
geometric_set]
Source: ISO 10303-42

#5: wireframe_with_topology

AIM element: [shell_based_wireframe_model
edge_based_wireframe_model]
Source: ISO 10303-42

### 5.1.11.6 Implicit_base_shape_representation

AIM element: shape_representation_with_parameters
Source: ISO 10303-520
Rule: 5.2.3.7.9
Reference path: shape_representation_with_parameters <=
shape_representation

 

#### 5.1.11.6.1 implicit_base_shape_representation to numeric_parameter (as base_shape_length)

AIM element: PATH
Reference path: shape_representation_with_parameters <=
shape_representation <=
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}

#### 5.1.11.6.2 implicit_base_shape_representation to orientation (as placement)

AIM element: PATH
Reference path: shape_representation_with_parameters <=
shape_representation <=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

### 5.1.11.7 Intermediate_shape

AIM element: shape_aspect_relationship
Source: ISO 10303-41
Reference path: shape_aspect_relationship
shape_aspect_relationship.name=’intermediate shape’

#### 5.1.11.7.1 as_is_shape

AIM element: shape_aspect
Source: ISO 10303-41
Reference path: shape_aspect_relationship
shape_aspect_relationship.name=’intermediate shape’
shape_aspect_relationship.relating_shape_aspect->
shape_aspect

### 5.1.11.7.2 to_be_shape

AIM element: shape_aspect
Source: ISO 10303-41
Reference path: shape_aspect_relationship
shape_aspect_relationship.name='intermediate shape'
shape_aspect_relationship.related_shape_aspect->
shape_aspect

## 5.1.11.8 Manufacturing_process_feature

#1 The feature is a Machining feature

AIM element: feature_definition
instanced_feature
Source: ISO 10303-520
Rule: 5.2.3.4.14, 5.2.3.7.3, 5.2.3.7.8
Reference path: [feature_definition<=
characterized_object]
[instanced_feature<=
shape_aspect]
(boss)
(flat_face)
(gear)
(marking)
(outer_round)
(pocket)
(protrusion)
(removal_volume)
(rib_top)
(round_hole)
(rounded_end)
(spherical_cap)
(step)
(slot)
(thread)

#2 The feature is a Transition feature

AIM element: feature_definition
instanced_feature
Source: ISO 10303-520
Reference path: transition_feature<=
shape_aspect
(fillet)
(edge_round)
(chamfer)

 

### 5.1.11.8.1 required_geometry

AIM element: advanced_brep_shape_representation
Source: ISO 10303-514
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

## 5.1.11.9 Model_element

AIM element: (geometric_representation_item)
(topological_representation_item)
Source: ISO 10303-42

### 5.1.11.9.1 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]

#2: faceted_b_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=>
faceted_brep]

#3: manifold_surface_with_topology

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
shell = closed_ shell)
(open_shell
shell = open_ shell)
shell<-
shell_based_surface_model. sbsm_boundary[i]
shell_based_ surface_model]

#4:non_topological_surface_and_wireframe

AIM element: PATH
Reference path: [geometric_ representation_ item = >
(surface<-)
(curve<-)
geometric_set.elements[i]
geometric_set = >
geometric_curve_set]

 

#5: wireframe_with_topology

AIM element: PATH
Reference path: [(geometric_ representation_ item =>
curve
edge_curve.edge_geometry
edge_curve <=
edge <-)
(topological_representation_item =>
edge <-)
(connected_edge_set.ces_edges[i]
connected_edge_set<-
edge_based_wireframe_model.ebwm_boundary[i]
edge_based_wireframe_model)
(oriented_edge.edge_element
oriented_edge<-
path.edge_list[i]
path=>
edge_loop<=
loop
loop<-
wire_shell.wire_shell_extent[i]
wire_shell
shell = wire_ shell
shell<-
shell_based_ wireframe _model. sbwm_boundary[i]
shell_based_wireframe_model)]

### 5.1.11.9.2 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

### 5.1.11.10 Ngon_base_shape

AIM element: ngon_shape_representation
Source: ISO 10303-240
Rule: 5.2.3.7.9
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation

### 5.1.11.10.1 circumscribed_or_across_flats

AIM element: representation_item.name
Source: ISO 10303-43
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
representation
representation.items[i] ->
representation_item
(representation_item.name = 'circumscribed diameter')
(representation_item.name = 'diameter across flats')

### 5.1.11.10.2 ngon_base_shape to numeric_parameter (as corner_radius)

AIM element: PATH
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'corner radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

### 5.1.11.10.3 ngon_base_shape to numeric_parameter (as diameter)

AIM element: PATH
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
representation
representation.items[i] ->
{representation_item
(representation_item.name = 'circumscribed diameter')
(representation_item.name = 'diameter across flats')}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

 

#### 5.1.11.10.4 ngon_base_shape to numeric_parameter (as number_of_sides)

AIM element: PATH
Reference path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation <=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of sides'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

### 5.1.11.11 Object_element_shape_representation

AIM element: shape_representation
Source: ISO 10303-41

#### 5.1.11.11.1 object_element_shape_representation to shape_representation_-type (as representation_type)

AIM element: IDENTICAL MAPPING
Source:

#### 5.1.11.11.2 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]

#2: faceted_b_rep

AIM element: PATH
Reference path: shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
[solid_model =>
manifold_solid_brep=>
faceted_brep]

#3: manifold_surface_with_topology

AIM element: PATH
Reference path: shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
[shell_based_surface_model]

#4:non_topological_surface_and_wireframe

AIM element: PATH
Reference path: shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
[shell_based_wireframe_model]
[edge_based_wireframe_model]

#5: wireframe_with_topology

AIM element: PATH
Reference path: shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
[geometric_set]
[geometric_curve_set]

## 5.1.11.12 Orientation

AIM element: placement
Source: ISO 10303-42

 

### 5.1.11.12.1 axis

AIM element: (axis1_placement.axis)
((axis2_placement_3d.axis)
(axis2_placement_3d.ref_direction))
Source: ISO 10303-42
ISO 10303-42
ISO 10303-42
Reference path: placement =>
(axis1_placement
axis1_placement.axis)
(axis2_placement_3d
(axis2_placement_3d.axis)
(axis2_placement_3d.ref_direction))

### 5.1.11.12.2 location

AIM element: placement.location
Source: ISO 10303-42

### 5.1.11.13 Part_placement

AIM element: mapped_item
Source: ISO 10303-43

### 5.1.11.13.1 part_placement to orientation (as resulting_orientation)

AIM element: PATH
Reference path: mapped_item
mapped_item.mapping_target->
representation_item=>
geometric_representation_item=>
placement=>
axis2_placement_3d

### 5.1.11.13.2 part_placement to orientation (as originating_orientation)

AIM element: PATH
Reference path: mapped_item
mapped_item.mapping_source->
representation_map
representation_map.mapping_origin->
representation_item=>
geometric_representation_item=>
placement=>
axis2_placement_3d

### 5.1.11.13.3 part_placement to part_shape (as oriented_physical_form)

AIM element: PATH
Reference path: mapped_item
mapped_item.mapping_source->
representation_map
representation_map.mapped_representation->
representation<-
{representation-=>
shape_representation}
property_definition_representation.used_representation
property_definition_representation
{property_definition_representation=>
shape_definition_representation}
property_definition_representation.definition->
represented_definition
represented_definition = property_definition
property_definition=>
product_definition_shape

### 5.1.11.14 Part_shape

AIM element: product_definition_shape
Source: ISO 10303-41

 

### 5.1.11.14.1 part_shape to base_shape (as base_shape_definition)

AIM element:
Source: ISO 10303-41
Rule: 5.2.3.5
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.relating_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

### 5.1.11.14.2 part_shape to object_element_shape_representation (as representation_form)

AIM element: PATH
Reference path: product_definition_shape <=
property_definition
represented_definition = property_definition
represented_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation

### 5.1.11.14.3 part_shape to shape_aspect (as elements)

AIM element: PATH
Reference path: product_definition_shape <-
shape_aspect.of_shape
shape_aspect

### 5.1.11.15 Shape_aspect

AIM element: shape_aspect
Source: ISO 10303-41

### 5.1.11.15.1 shape_aspect to model_element (as representation_shape)

AIM element: PATH
Source: ISO 10303-41
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
representation.items[i] ->
representation_item =>
(geometric_representation_item)
(topological_representation_item)

 

### 5.1.11.15.2 shape_aspect to shape_aspect_representation (as representation_-form)

#1: advanced_boundary_rep
#2: faceted_b_rep
#3: manifold_surface_with_topology
#4:non_topological_surface_and_wireframe
#5: wireframe_with_topology

AIM element:
Source: ISO 10303-41
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
{representation.items[i] ->
representation_item =>
(geometric_representation_item)
(topological_representation_item)}
representation =>
shape_representation=>
#1:[advanced_brep_shape_representation]
#2: [faceted_brep_shape_representation]
#3: [manifold_surface_shape_representation]
#4:[geometrically_bounded_wireframe_shape_representation
geometrically_bounded_surface_shape_representation]
#5: [edge_based_wireframe_shape_representation
shell_based_wireframe_shape_representation])

### 5.1.11.15.3 shape_aspect to manufacturing_process_feature (as element)

AIM element: IDENTICAL MAPPING

## 5.1.11.16 Shape_aspect_representation

AIM element: #1:[advanced_brep_shape_representation]
#2: [faceted_brep_shape_representation]
#3: [manifold_surface_shape_representation]
#4:[geometrically_bounded_wireframe_shape_representation
geometrically_bounded_surface_shape_representation
#5: [edge_based_wireframe_shape_representation
shell_based_wireframe_shape_representation]
Source: #1: ISO 10303-514
#2: ISO 10303-512
#3: ISO 10303-509
#4: ISO 10303-501, ISO 10303-502
#5:ISO 10303-510, ISO 10303-507
Reference path:

### 5.1.11.16.1 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]

#2: faceted_b_rep

AIM element: PATH
Reference path: [faceted_brep_shape_representation]
[solid_model =>
manifold_solid_brep=>
faceted_brep]

#3: manifold_surface_with_topology

AIM element: PATH
Reference path: [manifold_surface_shape_representation]
[shell_based_surface_model]

 

#4:non_topological_surface_and_wireframe

AIM element: PATH
Reference path: [geometrically_bounded_wireframe_shape_representation
geometrically_bounded_surface_shape_representation]
[shell_based_wireframe_model]
[edge_based_wireframe_model]

#5: wireframe_with_topology

AIM element: PATH
Reference path: [edge_based_wireframe_shape_representation
shell_based_wireframe_shape_representation]
(shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>)
[geometric_set]
[geometric_curve_set]

## 5.1.11.17 Shape_representation_type

AIM element: shape_representation
Source: ISO 10303-41

## 5.1.11.17.1 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 ]

#2: faceted_b_rep

AIM element: [faceted_brep_shape_representation]
Source: ISO 10303- 512
Reference path: [faceted_brep_shape_representation <=
shape_representation ]

#3: manifold_surface_with_topology

AIM element: [manifold_surface_shape_representation]

Source: ISO 10303- 509
Reference path: [manifold_surface_shape_representation <=
shape_representation ]

#4:non_topological_surface_and_wireframe

| | |
|---|---|
| AIM element: | [geometrically_bounded_wireframe_shape_representation geometrically_bounded_surface_shape_ representation] |
| Source: | ISO 10303-50<br>ISO 10303-502 |
| Reference path: | [(geometrically_bounded_wireframe_shape_representation <= shape_representation)<br>(geometrically_bounded_surface_shape_representation <= shape_representation)] |

#5: wireframe_with_topology

| | |
|---|---|
| AIM element: | [edge_based_wireframe_shape_representation<br>shell_based_wireframe_shape_representation] |
| Source: | ISO 10303-510<br>ISO 10303-507 |
| Reference path: | [(edge_based_wireframe_shape_representation <= shape_representation)<br>( shell_based_wireframe_shape_representation <= shape_representation ] |

## 5.2 AIM EXPRESS short listing

```
*)

SCHEMA process_planning_schema;

USE FROM aic_machining_feature;

USE FROM aic_advanced_brep;                                  -- ISO 10303-514

USE FROM aic_faceted_brep;                                   -- ISO 10303-512

USE FROM aic_manifold_surface;                               -- ISO 10303-509

USE FROM aic_edge_based_wireframe;                           -- ISO 10303-501

USE FROM aic_shell_based_wireframe;                          -- ISO 10303-502

USE FROM aic_geometrically_bounded_surface;                  -- ISO 10303-507

USE FROM aic_geometrically_bounded_wireframe;                -- ISO 10303-510

USE FROM aic_geometric_tolerances;                           -- ISO 10303-519

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,
   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,
   object_role);

USE FROM contract_schema                               -- ISO 10303-41
   (contract,
   contract_type);

USE FROM date_time_schema                              -- ISO 10303-41
   (calendar_date,
   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);

USE FROM external_reference_schema                     -- ISO 10303-41
   (external_source,
   externally_defined_item,
   externally_defined_item_relationship,
   pre_defined_item,
   source_item);

REFERENCE FROM geometry_schema                         -- ISO 10303-42
   (dummy_gri);
```

```
USE FROM geometry_schema                                    -- ISO 10303-42
   (axis1_placement,
   axis2_placement_3d,
   b_spline_curve_with_knots,
   bezier_curve,
   bezier_surface,
   cartesian_point,
   circle,
   elementary_surface,
   ellipse,
   geometric_representation_item,
   hyperbola,
   line,
   oriented_surface,
   parabola,
   placement,
   plane,
   point,
   quasi_uniform_curve,
   quasi_uniform_surface,
   surface);

USE FROM geometric_model_schema                             -- ISO 10303-42
   (edge_based_wireframe_model,
   faceted_brep,
   geometric_curve_set,
   geometric_set,
   manifold_solid_brep,
   shell_based_surface_model,
   shell_based_wireframe_model,
   solid_model);

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,
   contract_assignment,
   date_assignment,
   document_reference,
   document_usage_role,
   document_usage_constraint_assignment,
   effectivity_assignment,
   external_identification_assignment,
   group_assignment,
   identification_assignment,
   identification_role,
   organization_assignment,
   security_classification_assignment);

USE FROM material_property_definition_schema                -- ISO 10303-45
   (material_designation,
   material_property,
   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,
    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_schema                            -- ISO 10303-49
   (action_property,
   characterized_action_definition,
```

```
   characterized_resource_definition,
   action_resource_requirement,
   process_product_association,
   process_property_association,
   product_definition_process,
   property_process,
   requirement_for_action_resource,
   resource_property);

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_relationship,
   product_related_product_category);

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);

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
   (global_uncertainty_assigned_context,
   compound_representation_item,
```

 

```
   compound_item_definition,
   list_representation_item,
   set_representation_item,
   mapped_item,
   representation,
   representation_item,
   representation_map,
   value_representation_item);

USE FROM security_classification_schema
   (security_classification,
   security_classification_level);

USE FROM shape_dimension_schema
   (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
   (geometric_tolerance,
   limits_and_fits,
   plus_minus_tolerance,
   tolerance_value,
   tolerance_zone);

 REFERENCE FROM topology_schema                          -- ISO 10303-42
   (dummy_tri);


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);

USE FROM support_resource_schema                         -- ISO 10303-41
   (identifier);

REFERENCE FROM support_resource_schema                   -- ISO 10303-41
   (type_check_function);

(*
```

## 5.2.1 Fundamental concepts and assumptions

## 5.2.2 Process planning schema types

### 5.2.2.1 action_item

A **action_item** identifies an **feature_dependency**, or **product_definition_formation** to which a referenced **action** may be assigned.

EXPRESS specification:

```
*)
TYPE action_item = SELECT (
   feature_dependency,
   product_definition_formation);
END_TYPE;
(*
```

### 5.2.2.2 action_method_item

A **action_method_item** identifies an **externally_defined_schema** to which a referenced **action_method** may be assigned.

EXPRESS specification:

```
*)
TYPE action_method_item = SELECT (
   externally_defined_schema,
   property_definition);
END_TYPE;
(*
```

### 5.2.2.3 action_request_item

A **action_request_item** identifies an **fixture_contract**, **tool_contract**, or **product_definition_formation** to which a referenced **action_request** may be assigned.

EXPRESS specification:

```
*)
TYPE action_request_item = SELECT (
   product_definition_formation,
   fixture_contract,
   tool_contract);
END_TYPE;
(*
```

 

### 5.2.2.4 approval_item

A **approval_item** identifies an **revision**, to which a referenced **approval** may be assigned.

EXPRESS specification:

```
*)
TYPE approval_item = SELECT (
   revision);
END_TYPE;
(*
```

### 5.2.2.5 classification_item

A **classification_item** identifies an **externally_defined_representation_with_parameters** that may be assigned.

EXPRESS specification:

```
*)
TYPE classification_item = SELECT (
   externallY_defined_representation_with_parameters);
END_TYPE;

(*
```

### 5.2.2.6 contract_item

A **contract_item** identifies an **process_plan_specification, document_file, fixture_assembly, product_definition_formation, in_facility_location** or **tool_assembly** to which a referenced **contract** may be assigned.

EXPRESS specification:

```
*)
TYPE contract_item = SELECT (
   process_plan_specification,
   document_file,
   fixture_assembly,
   product_definition_formation,
   tool_assembly,
   in_facility_location);
END_TYPE;
(*
```

### 5.2.2.7 date_item

A **date_item** identifies an **process_plan_security** or **versioned_action_request** to which a referenced **date** may be assigned.

EXPRESS specification:

```
*)
TYPE date_item = SELECT (
   process_plan_security,
   versioned_action_request);
END_TYPE;
(*
```

### 5.2.2.8 document_reference_item

A **document_reference_item** identifies an **action_method, action_resource_requirement, controller, date_and_time, externally_defined_schema, fixture_assembly, fixture_assembly_element, machine_usage, manufacturing_activity, process_plan_activity, product_definition, product_definition_formation, property_definition, tool_assembly, tool_assembly_element, tool_placement_instruction**, **externally_defined_dimension_definition** or **workstation** to which a referenced **document** may be assigned.

EXPRESS specification:

```
*)
TYPE document_reference_item = SELECT (
   action_method,
   action_resource_requirement,
   controller,
   date_and_time,
   externally_defined_schema,
   externally_defined_dimension_definition,
   fixture_assembly,
   fixture_assembly_element,
   machine_usage,
   Manufacturing_activity,
   process_plan_activity,
   product_definition,
   product_definition_formation,
   property_definition,
   manufacturing_process,
   tool_assembly,
   tool_assembly_element,
   tool_placement_instruction,
   workstation);
END_TYPE;

(*
```

### 5.2.2.9 document_usage_constraint_item

A **document_usage_constraint_item** identifies an **illustration** which may be assigned.

EXPRESS specification:

```
*)
TYPE document_usage_constraint_item = SELECT (
   illustration);
END_TYPE;
(*
```

 

### 5.2.2.10 effectivity_item

A **effectivity_item** identifies a **product_definiton_formation** to which a referenced **effectivity** may be assigned.

EXPRESS specification:

```
*)
TYPE effectivity_item = SELECT (
   product_definition_formation);
END_TYPE;
(*
```

### 5.2.2.11 external_identification_item

A **external_identification_item** identifies an **document**, **externally_defined_class** or **externally_-defined_general_property** to which a referenced **external_identification** may be assigned.

EXPRESS specification:

```
*)
TYPE external_identification_item = SELECT (
   document,
   externally_defined_class,
   externally_defined_general_property);
END_TYPE;
(*
```

### 5.2.2.12 group_item

A **group_item** identifies a **workstation**, **feature_process** or **instanced_feature** to which a referenced **group** may be assigned.

EXPRESS specification:

```
*)
TYPE group_item = SELECT (
   workstation,
   instanced_feature,
   feature_process);
END_TYPE;
(*
```

### 5.2.2.13 identification_assignment_item

A **identification_assignment_item** identifies an **document_file** to which a referenced **identification_role** may be assigned.

EXPRESS specification:

```
*)
TYPE identification_assignment_item= SELECT (
   document_file);
END_TYPE;
(*
```

### 5.2.2.14 organization_item

EXPRESS specification:

An **organization_item** identifies an **action, process_plan_activity, fixture_assembly_element, known_source, machine, part_contract, product_definition,** or **tool_assembly_element** to which a referenced **organization** may be assigned.

```
*)
TYPE organization_item = SELECT (
   process_plan_activity,
   fixture_assembly_element,
   known_source,
   machine,
   part_contract,
   product_definition,
   tool_assembly_element);
END_TYPE;
(*
```

### 5.2.2.15 security_classification_item

A **organization_item** identifies an **process_plan_activity** or **process_plan_version** to which a referenced **security_classification_item** may be assigned.

EXPRESS specification:

```
*)
TYPE security_classification_item = SELECT (
   process_plan_activity,
   process_plan_version);
END_TYPE;
(*
```

## 5.2.3 Process planning schema entities

### 5.2.3.1 Process planning schema entity definitions

 

### 5.2.3.1.1 allowed_time

An **allowed_time** is an **action_property** that specifies an amount of time for the completion of an activity.

EXPRESS specification:

```
*)
ENTITY allowed_time
   SUBTYPE OF (action_property);
WHERE

     (* allowance_factor *)

  wr1: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
     (SIZEOF(QUERY ( it <* apr.representation.items |
     ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF
     (it)) = 2) AND (it.name = 'allowance factor'))))=1)))=0));

    (* standard_time *)

   wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
      (SIZEOF(QUERY ( it <* apr.representation.items |
      ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(
      it)) = 2) AND (it.name = 'standard time')) )) = 1) )) = 0);

     (* allowed type *)

  wr3: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
     (SIZEOF(QUERY ( it <* apr.representation.items |
     (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'allowed type'))))<=1)))=0));

     (* Performance_rate.source *)

  wr4: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
    (SIZEOF(QUERY ( it <* apr.representation.items |
    (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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'.

## 5.2.3.1.2 alternate_action_method_relationship

An **alternate_action_method_relationship** is an **action_method_relationship** which relates an activity to an alternate activity.

EXPRESS specification:

```
*)
ENTITY alternate_action_method_relationship
   SUBTYPE OF (action_method_relationship);
WHERE

   wr1: ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY'
      IN TYPEOF(SELF.related_method));


   wr2: ('PROCESS_PLANNING_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**.

## 5.2.3.1.3 alternate_plan_relationship

An **alternate_plan_relationship** is an **action_relatonship** that relates a process plan with an alternate process plan.

 

EXPRESS specification:

```
*)
ENTITY alternate_plan_relationship
   SUBTYPE OF (action_relationship);
WHERE
 wr1: ((('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION')
    IN TYPEOF (SELF.relating_action))AND
    (('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION')
    IN TYPEOF (SELF.related_action)));

 wr2: (SIZEOF(QUERY ( edir <* USEDIN(SELF.related_action,
    'PROCESS_PLANNING_SCHEMA.ID_ATTRIBUTE.IDENTIFIED_ITEM') |
    ((edir.attribute_value='alternate process plan'))))=1);

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **relating_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.

### 5.2.3.1.4 ancillary_activity

An **ancillary_activity** is a type of **manufacturing_activity** which is not covered by any other category of activity

EXPRESS specification:

```
*)
ENTITY ancillary_activity
   SUBTYPE OF (manufacturing_activity);

END_ENTITY;
(*
```

### 5.2.3.1.5 ancillary_setup

An **ancillary_setup** is an **action_method** which is not covered by any other category of setup activity.

EXPRESS specification:

```
*)
ENTITY ancillary_setup
   SUBTYPE OF (process_plan_activity);

END_ENTITY;
(*
```

### 5.2.3.1.6 applied_action_request_assignment

A **applied_action_request_assignment** specifies those **action_request_item**s for which a design exception has been identified.

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_item**s for which a particular **action_request** is applicable.

### 5.2.3.1.7 applied_action_assignment

A **applied_action_assignment** specifies those **action_item**s to which an **action** 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_item**s for which a particular **action** is applicable.

### 5.2.3.1.8 applied_action_method_assignment

A **applied_action_method_assignment** specifies those **approved_method_item**s 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_methd_item**s for which a particular **action_method** is applicable.

 

### 5.2.3.1.9 applied_approval_assignment

A **applied_approval_assignment** specifies those **approved_item**s to which an **approval** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_approval_assignment
   SUBTYPE OF (approval_assignment);
   items: SET[1:?] OF approval_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **approved_item**s to which an **approval** is assigned.

### 5.2.3.1.10 applied_classification_assignment

A **applied_classification_assignment** specifies those **classification_item**s 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_item** s to which an **classification_role** is assigned.

### 5.2.3.2 applied_library_assignment

A **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.

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
      ('PROCESS_PLANNING_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,
 'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATED_ITEM')|
     ((edir.name = 'name scope') AND
     ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_GENERAL_PROPERTY'
     IN TYPEOF(edir.relating_item))AND
     (SIZEOF(QUERY ( gpa <* USEDIN(edir.relating_item,
    'PROCESS_PLANNING_SCHEMA.GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION')|
     ((gpa.name = 'definitional') AND
     (SIZEOF (QUERY (pdr <* USEDIN( (QUERY(pd <*
     gpa.derived_definition.derived_property_select |
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(pd))),
 'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ((pdr.used_representation.name = 'property value')AND
     (SIZEOF(QUERY(ri <* pdr.used_representation.items|
     'PROCESS_PLANNING_SCHEMA.VALUE_REPRESENTATION_ITEM'
     IN TYPEOF(ri))) >=1)))) =1)))) =1)))) =1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **role** attribute shall define a **classificatioin_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'

## 5.2.3.2.1 applied_contract_assignment

A **applied_contract_assignment** specifies those **contract_item**s to which an **contract** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_contract_assignment
   SUBTYPE OF (contract_assignment);
   items: SET[1:?] OF contract_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **contract_item**s to which an **contract** is assigned.

 

### 5.2.3.2.2 applied_date_assignment

A **applied_date_assignment** specifies those **date_item**s 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_item**s to which an **date**is assigned.

### 5.2.3.2.3 applied_document_reference

A **applied_document_assignment** specifies those **document_reference_item**s 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_item**s to which an **document** is assigned.

### 5.2.3.2.4 applied_document_usage_constraint_assignment

A **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_usage_constraint_item;
END_ENTITY;
(*
```

Attribute definition:

**items:** the set of **document_usage_constraint_item**s to which an **document** is assigned.

### 5.2.3.2.5 applied_effectivity_assignment

A **applied_effectivity_assignment** specifies those **effectivity_item**s to which an **effectivitty** 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_item**s to which an **effectivity** is assigned.

### 5.2.3.2.6 applied_external_identification_assignment

A **applied_external_identification_assignment** specifies those **external_identification_item**s 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_item**s to which an **external_identification** is assigned.

### 5.2.3.2.7 applied_group_assignment

A **applied_group_assignment** specifies those **group_item**s 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_item** s to which an **group** is assigned.

### 5.2.3.2.8 applied_identification_assignment

A **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_item** to which an **identification_role** is assigned.

### 5.2.3.2.9 applied_organization_assignment

A **applied_organization_assignment** specifies those **organization_item**s 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_item**s to which an **organization** is assigned.

### 5.2.3.2.10 applied_security_classification_assignment

A **applied_security_classification_assignment** specifies those **security_classification_item**s 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_item**s to which an **security_classification** is assigned.

## 5.2.3.2.11 block_shape_representation

A **block_shape_representation** specifies the representation of a shape that is a rectangular volume defined as a rectangular area of a defined length. The enclosed area is defined by four straight sides with opposite sides equal in length.

EXPRESS specification:

```
*)
 ENTITY block_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
   wr1: (SIZEOF(SELF.items) = 4);
   wr2: (SIZEOF(QUERY ( it <* SELF.items |
      (('PROCESS_PLANNING_SCHEMA.PLACEMENT'
      IN TYPEOF(it)) AND (it.name = 'orientation')))) = 1);
   wr3: (SIZEOF(QUERY ( it <* SELF.items |
      ((SIZEOF(
      ['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1);
   wr4: (SIZEOF(QUERY ( it <* SELF.items |
      ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'width')))) = 1);
   wr5: (SIZEOF(QUERY ( it <* SELF.items |
      ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'height')))) = 1);
 END_ENTITY; -- block_shape_representation
(*
```

Formal propositions:

**WR1:** The **block_shape_representation** shall contain exactly four **representation_item**s in its set of **items**.

**WR2:** One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

**WR3:** One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

**WR4:** One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a

 

**name** of 'width'.

**WR5:** One of the **representation_item**s used for the implicit representation of a **block_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'height'.

Informal propositions:

**IP1:** The **block_shape_representation** shall be defined at the center of the rectangular area in the X-Y plane with the width of the rectangle in the X direction, the height of the rectangle in the Y direction, and the length of the rectangle in the Z direction.

## 5.2.3.2.12 part_holding_position

The **part_holding_position** is a type of **resource_property** that defines the positioning of a clamp to hold down the piece being manufactured.

EXPRESS specification:

```
*)
ENTITY part_holding_position
   SUBTYPE OF (resource_property);
WHERE

  wr1: SIZEOF(QUERY ( pdp_imp <* QUERY ( pdp <* SELF.resource.operations |
     ('PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(pdp)) ) |
     (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
     (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
     IN TYPEOF(ppa.property_or_shape)) AND
     (ppa.description = 'part holding mating shape')) )) = 0) )) >= 0;


   wr2: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
      (SIZEOF(QUERY ( ap <* apr.representation.items |
      ((('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(ap)) AND
      (ap.name = 'part holding position type')) AND
      (ap.description IN ['jack','locator','clamp'])))) = 1))) = 0));

   wr3: ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(SELF.resource)) AND
      (SIZEOF(QUERY(rfar <* SELF.resource.resources |
      ('PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY'
      IN TYPEOF(rfar))))=1);

   wr4: (NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
      (SIZEOF(QUERY ( ap <* apr.representation.items |
      (('PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(ap)) AND
      (ap.name = 'part holding location')) )) <= 1) )) = 0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **part_holding_position** shall reference through the **resource** attribute a **action_resource_-requirement** that shall reference through the **operations** attribute a **property_process** that is referenced through the **process** attribute by a **process_property_association** with a description of 'part holding mating shape' that references through the **property_or_shape** attribute a **shape_aspect** with **description** of 'clamp mating shape'.

**WR2:** The **part_holding_position** shall be referenced through the **property** attribute by a **resource_-property_representation** that references through the **representation** attribute **discriptive_-representation_item** with a **name** of 'position type' and **description** of either 'jack', 'locator',or 'clamp'.

**WR3:** The **part_holding_position** shall reference through the **resource** attribute **fixture_assembly**

**WR4:** The **part_holding_position** shall be referenced through the **property** attribute by a **resource_-property_representation** that references through the **representation** attribute **cartesian_point** with a **name** of 'part holding location'.

## 5.2.3.2.13 class

A **class** is a type of **group** that specifies a type of classification assignment.

EXPRESS specification:

```
*)
ENTITY class
   SUBTYPE OF (group);
END_ENTITY;
(*
```

## 5.2.3.2.14 continuous_process_relationship

The **continuous_process_relationship** is a type of **sequential_method** that relates a **manufacturing_-process** to a **manufacturing_process**.

EXPRESS specification:

```
*)
ENTITY continuous_process_relationship

   SUBTYPE OF (sequential_method);

WHERE

 wr1: ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
   IN TYPEOF(SELF.related_method));
```

 

```
 wr2:   ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
   IN TYPEOF(SELF.relating_method));

  wr3: (SELF.description IN ['serial','batch', 'serial and batch']);

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **related_method** attribute shall be **manufacturing_process**.

**WR2:** The **relating_method** attribute shall be **manufacturing_process**.

**WR3:** The des**cription shall be one of** 'serial','batch', or 'serial and batch'.

## 5.2.3.2.15 controller

A **controller** is a type of **action_resource** that is the computerized numerical controller that supplies input to a **machine**.

EXPRESS specification:

```
*)
ENTITY controller
   SUBTYPE OF (action_resource);

WHERE

 (* controller to specification *)

   wr1: SIZEOF(QUERY (adr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
      ('PROCESS_PLANNING_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.

## 5.2.3.2.16 controller_program

A **controller_program** is a type of **document** that defines instructions for machining a part or a portion of a part.

EXPRESS specification:

```
*)
ENTITY controller_program
   SUBTYPE OF (document);
WHERE

  (* controller_program.revision *)

   wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')|
     (duc.subject_element = 'controller program revision level'))) = 1;

  (* controller_program.validation data and time *)

   wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
      ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))) |
      (SIZEOF(QUERY ( d <* adr.items |
      ((('PROCESS_PLANNING_SCHEMA.DATE_AND_TIME' IN TYPEOF(d)) AND
      ('PROCESS_PLANNING_SCHEMA.DATE' IN TYPEOF(d.date_component))) AND
      ('PROCESS_PLANNING_SCHEMA.LOCAL_TIME' IN TYPEOF(d.time_component)))
     )) = 1))) = 1;

  (* controller_program to manufacturing_activity *)

   wr3: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
      ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))) | (SIZEOF(QUERY ( d <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY' IN TYPEOF(d))))
      >= 1))) = 1;

 (* controller_program to part_version *)

   wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
      (('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(adr)) AND
      (NOT (SIZEOF(QUERY ( pdf <* adr.items|
      (('PROCESS_PLANNING_SCHEMA.'+
      'PRODUCT_DEFINITION_FORMATION'
      IN TYPEOF(pdf)) AND
      (NOT (SIZEOF(QUERY ( pd <* USEDIN(pdf ,
      'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION.FORMATION') |
      (pd.frame_of_reference.name = 'part definition'))) = 1)))))
      = 1))))) = 0;

  (* controller_program to tool_assembly *)

   wr5: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
      ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))) | (SIZEOF(QUERY ( d <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(d))
     )) >= 1))) = 1;
```

 

```
   wr6: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
      ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))) | (SIZEOF(QUERY ( d <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(d))))
      >= 1))) = 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** A **controller_program** shall be referenced by exactly one **document_usage_constraint** with **name** of 'controller program revision level'.

**WR2:** The **controller_program** shall be the **assigned_document** in exactly one **applied_document_reference** that contains exactly one **date_and_time** in its set of **items** with a **date_component** of **date** and a **time_component** of **local_time**.

**WR3:** The **controller_program** shall be the **assigned_document** in exactly one **applied_document_reference** that contains exactly one **manufacturing_activity**.

**WR4:** The **controller_program** shall be the **assigned_document** in exactly one **applied_document_reference** that contains exactly one **product_definition_formation** that requires a **product_definition_context** with a **name** of 'part definition'

**WR5:** The **controller_program** shall be the **assigned_document** in exactly one **applied_document_reference** that contains exactly one **tool_assembly**.

**WR6:** The **controller_program** shall be the **assigned_document** in exactly one **applied_document_reference** that contains exactly one **manufacturing_process.**

## 5.2.3.2.17 cylindrical_shape_representation

A **cylindrical_shape_representation** specifies representation of a shape that is a cylindrical volume defined as a circular area of a defined length. The enclosed area is defined by a circle with a specified radius.

EXPRESS specification:

```
*)
 ENTITY cylindrical_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
   wr1: (SIZEOF(SELF.items) = 3);
   wr2: (SIZEOF(QUERY ( it <* SELF.items | ((
      'PROCESS_PLANNING_SCHEMA.PLACEMENT'
      IN TYPEOF(it)) AND (it.name = 'orientation')))) = 1);
   wr3: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1);
   wr4: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
```

```
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'diameter')))) = 1);
 END_ENTITY; -- cylindrical_shape_representation
(*
```

Formal propositions:

**WR1:** The **cylindrical_shape_representation** shall contain exactly three **representation_item**s in its set of **items**.

**WR2:** One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

**WR3:** One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

**WR4:** One of the **representation_item**s used for the implicit representation of a **cylindrical_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'diameter'.

Informal propositions:

**IP1:** The location of the **cylindrical_shape_representation** shall be defined to be at the center of the circle that defines the cylinder.

**IP2**: The **cylindrical_shape_representation** shall be defined by forming a circular profile in the X-Y plane, and the length along the z direction.

## 5.2.3.2.18 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.

EXPRESS specification:

```
*)
ENTITY defining_action_method_relationship
   SUBTYPE OF (action_method_relationship);
WHERE
   wr1: ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY'
      IN TYPEOF(SELF.related_method));
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **related_method** shall be a **process_plan_activity**.

 

## 5.2.3.2.19 directed_dimensional_location

A **directed_dimension_location** specifies is a type of **dimension_location** that identifies the direction to measure the location dimension.

EXPRESS specification:

```
*)
 ENTITY directed_dimensional_location
  SUBTYPE OF (dimensional_location);
 END_ENTITY; -- directed_dimensional_location
(*
```

Attribute definitions:

**SELF\shape_aspect_relationship.relating_shape aspect:** the origin of the directed dimension.

**SELF\shape_aspect_relationship.related_shape_aspect:** the target of the directed dimension.

## 5.2.3.2.20 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 process planning specifications.

EXPRESS specification:

```
*)
ENTITY document_file
   SUBTYPE OF (document,characterized_object);
WHERE

   wr1 : SIZEOF(QUERY ( drt <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT')|
      (drt.name IN ['physical','digital']))) = 1;

(* document_file.version_id *)

   wr2 : SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS')|
      (('PROCESS_PLANNING_SCHEMA.IDENTIFICATION_ASSIGNMENT'
      IN TYPEOF(adr)) AND (adr.role.name = 'version')))) <= 1;

(* document_file to external_file_information *)

   wr3 : SIZEOF(QUERY ( aeia <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')|
      ('PROCESS_PLANNING_SCHEMA.EXTERNAL_IDENTIFICATION_ASSIGNMENT'
      IN TYPEOF(aeia)))) >= 0;

(* document_file to document_file_properties *)

 wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')|
    ((pd.name='document property') AND (SIZEOF(QUERY ( pdr <*
    USEDIN(pd,
  'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
    (('PROCESS_PLANNING_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'.

## 5.2.3.2.21 document_file_properties

The **document_file_properties** is a type of **representation** that defines properties for **document_file**.

EXPRESS specification:

```
*)
 ENTITY document_file_properties
   SUBTYPE OF (representation);
 WHERE

  wr1: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='country code') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr2: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='detail level') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr3: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='geometry type') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr4: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='language code') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr5: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='creating interface') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
```

 

```
    IN TYPEOF (r))))) <=1;

  wr6: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='creating operating system') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr7: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='creating system') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr8: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='data format') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr9: SIZEOF(QUERY(r <* SELF.items |
    ((r.name='format character code') AND
    ( 'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF (r))))) <=1;

  wr10: SIZEOF(QUERY(r <* SELF.items |
     ((r.name='file size') AND
     (SIZEOF(['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
     TYPEOF(r)) = 2)))) <=1;

  wr11: SIZEOF(QUERY(r <* SELF.items |
     ((r.name='page count') AND
     (SIZEOF(['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.COUNT_MEASURE'] *
     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'.

## 5.2.3.2.22 design_reference

A **design_reference** is a type of **document** that is a graphics representation of a part.

EXPRESS specification:

```
*)
ENTITY design_reference
   SUBTYPE OF (document);
WHERE

  (* design_reference.revision_level *)

   wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')|
      (duc.subject_element = 'drawing revision level'))) = 1;

 (* design_reference to process_plan_activity*)


 wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
   ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
   IN TYPEOF(dr))) |
   (NOT(SIZEOF(QUERY ( d <* adr.items |
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d))
   )) >= 1)))) =0;


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**

## 5.2.3.2.23 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.

EXPRESS specification:

```
*)
ENTITY externally_defined_class
   SUBTYPE OF (externally_defined_item,class);
WHERE
 wr1: 'PROCESS_PLANNING_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,
    'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
    (aoa.role.name = 'library supplier'))) = 1;

   (* class_BSU.version *)

 wr4: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')|
    (('PROCESS_PLANNING_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.

## 5.2.3.2.24 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.

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,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (adr.assigned_document.description =
   'externally defined 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'.

## 5.2.3.2.25 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.

EXPRESS specification:

```
*)
ENTITY externally_defined_general_property
   SUBTYPE OF (general_property, externally_defined_item);
WHERE
 wr1: 'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);
 wr2: SELF.source.name = 'ISO 13584 library';

    (* property_BSU.version *)
```

 

```
 wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')|
    (('PROCESS_PLANNING_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
    IN TYPEOF(aoa)) AND
    (aoa.role.name = 'property version')))) = 1;

 wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM')|
    ((ap.name = 'name scope') AND
    ('PROCESS_PLANNING_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**.

## 5.2.3.2.26 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.

EXPRESS specification:

```
*)
ENTITY externally_defined_representation_with_parameters
   SUBTYPE OF (representation);
WHERE

 (* externally_defined_representation_with_parameters
 to library_part_assignment *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')|
    (('PROCESS_PLANNING_SCHEMA.APPLIED_LIBRARY_ASSIGNMENT'
  IN TYPEOF(adr)) AND (adr.role.name='definitional class membership'))))= 1;

 (* externally_defined_representation_with_parameters.placement*)

 wr2: SIZEOF(QUERY ( adr <* SELF.items |
    ('PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(adr)))) <= 1;

 (* externally_defined_representation_with_parameters.location *)
```

```
 wr3: SIZEOF(QUERY ( adr <* SELF.items |
    (('PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(adr)) AND
    ('PROCESS_PLANNING_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**.

## 5.2.3.2.27 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.

EXPRESS specification:

```
*)

ENTITY externally_defined_schema
    SUBTYPE OF (externally_defined_item);
WHERE

 wr1: 'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);
 wr2: SELF.source.name = 'ISO 10303 part';

 wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (adr.assigned_document.description ='externally defined schema')
   )) <= 1;

 wr4: (SELF.item_id IN ['externally defined schema','executable']);

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 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'.

 

## 5.2.3.2.28 feature_identification_item

The **feature_identification_item** is a type of **class**, **feature_identification_item** identifies a feature in the feature classification structure.

EXPRESS specification:

```
*)
ENTITY feature_identification_item
  SUBTYPE OF (class);
WHERE

 (* feature_identification_item.feature *)

   wr1: (NOT(SIZEOF(QUERY ( adr <* QUERY ( ga <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.GROUP_ASSIGNMENT.ASSIGNED_GROUP')|
   ('PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT' IN TYPEOF(ga)) ) |
      (SIZEOF(QUERY ( ins <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.INSTANCED_FEATURE' IN TYPEOF(ins)) ))
      = 1))) = 0));

   wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.GROUP_RELATIONSHIP.RELATING_GROUP')
        | ('PROCESS_PLANNING_SCHEMA.FEATURE_INTERACTION' IN TYPEOF(
        adr)) )) >= 0;

   wr3: (NOT(SIZEOF(QUERY ( adr <* QUERY ( ga <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.GROUP_ASSIGNMENT.ASSIGNED_GROUP')|
   ('PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT' IN TYPEOF(ga)) ) |
      (SIZEOF(QUERY ( ins <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.FEATURE_PROCESS' IN TYPEOF(ins)) ))
      <= 1))) = 0));


END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **feature_identification_item** shall be referenced by exactly one **applied_group_assignment** of kind **group_assignment** through the **assigned_group** that references exactly one **instanced_feature** in the list of **items**.

**WR2:** The **feature_identification_item** shall be referenced by zero or more **feature_interaction**of kind **group_relationship** through the **relating_group**.

**WR3:** The **feature_identification_item** shall be referenced by exactly one **applied_group_assignment** of kind **group_assignment** through the **assigned_group** that references exactly one **feature_process** in the list of **items**.

## 5.2.3.2.29 feature_interaction

The **feature_interaction** is a type of **group_relationship** that **related_group** is the previous **feature_-identification_item** and the **relating_group** is the following **feature_identification_item**.

EXPRESS specification:

```
*)
ENTITY feature_interaction
   SUBTYPE OF (group_relationship);
WHERE
   wr1: ('PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM'
      IN TYPEOF(SELF.related_group));
   wr2: ('PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM'
      IN TYPEOF(SELF.relating_group));
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **feature_identification_item** shall be reference through the **related_group**.

**WR2:** The **feature_identification_item** shall be reference through the **relating_group**.

## 5.2.3.2.30 feature_dependency

The **feature_dependency** is a type of **group_relationship** that defines the classification structure of features used by process planning activity.

EXPRESS specification:

```
*)
ENTITY feature_dependency
   SUBTYPE OF (group_relationship);
WHERE

 (* feature_dependency or feature_identification_item*)

   wr1: ('PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM'
      IN TYPEOF(SELF.related_group));
   wr2: ('PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM'
      IN TYPEOF(SELF.relating_group));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **feature_dependency** shall reference through the **related_group** attribute a **feature_-identification_item**.

**WR2:** The **feature_dependency** shall reference through the **relating_group** attribute a **feature_-identification_item**.

## 5.2.3.2.31 Feature_process

The **feature_process** is a type of **action_method** that defines the process information being applied to a feature.

 

EXPRESS specification:

```
*)
ENTITY feature_process
  SUBTYPE OF (action_method);

WHERE
   wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')|
      (('PROCESS_PLANNING_SCHEMA.SEQUENTIAL_METHOD' IN TYPEOF(sar)) AND
      ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY'
      IN TYPEOF(sar.related_method))) )) >= 1;

   wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')|
     (('PROCESS_PLANNING_SCHEMA.SERIAL_ACTION_METHOD' IN TYPEOF(sar)) AND
      ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
      IN TYPEOF(sar.related_method))) )) = 1;

   wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
      (NOT(SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
      (NOT(SIZEOF(QUERY ( it <* apr.representation.items |
      (( SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
      TYPEOF(it)) = 1) AND
      (it.name = 'process parameter')) ))
      >= 0))     )) = 0))     )) = 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 **sequential_method** that references through the **related_method** an **manufacturing_activity**.

**WR2:** There shall be one or more references through the **relating_method** attribute of an **action_-method_relationship** of type **serial_method** that references through the **related_method** an **manufacturing_process**.

**WR3:** There shall be zero, one, or many reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **representation_item** with a **name** of 'property parameter', and the **representation_item** shall be one of **descriptive_representation_item**, or **measure_representation_-item**.

## 5.2.3.2.32 fixture_assembly

The **fixture_assembly** is a type of **action_resource** that is used to support the part.

EXPRESS specification:

```
*)
ENTITY fixture_assembly
   SUBTYPE OF (action_resource);
WHERE

 (* fixture_assembly.assembly_instruction *)

 wr1: SIZEOF(QUERY ( adr <* SELF.usage | ((
   'PROCESS_PLANNING_SCHEMA.ACTION_METHOD' IN TYPEOF(adr)) AND
   (adr.name = 'assembly instruction')))) = 1;

 (* fixture_assembly.configuration *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (adr.description = 'configuration'))) = 1;

 (* fixture_assembly to fixture_assembly_element *)
 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATING_RESOURCE')|
   (('PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_RELATIONSHIP'
   IN TYPEOF(ap)) AND
   (SIZEOF([
   'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
   'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT'] *
    TYPEOF(ap.related_resource)) =1)
   ))) >= 1;

 (* fixture_assembly to fixture_contract *)

 wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT.ITEMS')|
   ('PROCESS_PLANNING_SCHEMA.FIXTURE_CONTRACT'
   IN TYPEOF(adr.assigned_contract)))) >= 1;


 (* fixture_assembly to design_reference *)

 wr5: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE'
   IN TYPEOF(adr.assigned_document)))))<=1;


 (* fixture_assembly to shape_aspect *)


 wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')|
   ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT'
   IN TYPEOF(ar))) | (NOT (SIZEOF(QUERY ( pdp_imp <*
   QUERY ( pdp <* arr.OPERATIONS |
   ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS'
   IN TYPEOF(pdp))) | (SIZEOF(QUERY ( ppa <*
   USEDIN(pdp_imp,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
   ((('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
   IN TYPEOF(ppa.property_or_shape)) AND
```

 

```
   (ppa.description = 'fixture assembly shape'))))) =
   0))) = 0)))) >= 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **fixture_assembly** shall reference through **usage** an **action_method** with a **name** of 'assembly instruction'

**WR2:** There shall be exactly one reference from a **resource_property** through **resourece** with a **description** of 'configuration'.

**WR3:** There shall be one or more references from a **action_resource_relationship** of type **fixture_-assembly_relationship** through **relating_resource** that references a **fixture_assembly** through **related_-resource**.

**WR4:** There shall one or more references from a **contract_assignment** of kind **applied_contract_-assignment** through the **items** attribute that references a **fixture_contract** through the **assigned_contract**.

**WR5:** There shall one or more references from an **applied_document_reference** through the **items** attribute that references a **design_reference** through the **assigned_contract**.

**WR6:** There shall be a reference through the **property** attribute a **resource_property** that shall reference through the **resource** attribute a **action_resource_requirement** that shall reference through the **operations** attribute a **property_process** that is referenced through the **process** attribute by a **process_-property_association** with a description of 'fixture assembly shape' that references through the **property_-or_shape** attribute a **shape_aspect**.

## 5.2.3.2.33 fixture_assembly_element

The **fixture_assembly_element** is a type of **action_resource** that is used to define portions of the **fixture_assembly**.

EXPRESS specification:

```
*)
ENTITY fixture_assembly_element
   SUBTYPE OF (action_resource);
WHERE

 (* fixture_assembly_element.catalogue_number *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.DOCUMENT'
   IN TYPEOF(adr.assigned_document)) AND
   (adr.assigned_document.description ='fixture catalogue number')
   ))) <= 1;

 (* fixture_assembly_element.company_name *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
```

```
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
   (adr.assigned_organization.description = 'fixture company name')
   )) = 1;

 (* fixture_assembly_element.id *)

 wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (adr.description = 'fixture element identification'))) = 1;

 (* fixture_assembly_element.weight *)

 wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (NOT (SIZEOF(QUERY ( rpr <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MASS_MEASURE_WITH_UNIT'
   IN TYPEOF(rpr)) AND (rpr.name = 'fixture element weight'))
   )) <= 1)))) = 0)))) = 0;

 (* fixture_assembly_element to shape_aspect *)

 wr5: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')|
   ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT'
   IN TYPEOF(ar))) | (NOT (SIZEOF(QUERY ( pdp_imp <*
   QUERY ( pdp <* arr.OPERATIONS |
   ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS'
   IN TYPEOF(pdp))) | (SIZEOF(QUERY ( ppa <*
   USEDIN(pdp_imp,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
   ((('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.description = 'fixture assembly element shape'))))) =
   0))) = 0)))) >= 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall exactly zero or one references from a **applied_document_assignment** through the **items** attribute that references a **document** with a **description** of 'fixture catalogue number' of through the **assigned_document**.

**WR2:** There shall exactly one references from a **applied_organization_assignment** through the **items** attribute that references a **organization** with a **description** of 'fixture company name' of through the **assigned_organization**.

**WR3:** There shall be exactly one reference from a **resource_property** through **resource** with a **description** of 'fixture element identification'.

 

**WR4:** There shall be exactly zero or one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **mass_-measure_with_unit** with a **name** of 'fixture element weight' through the list of **items**.

**WR5:** There shall be a reference through the **property** attribute a **resource_property** that shall reference through the **resource** attribute a **action_resource_requirement** that shall reference through the **operations** attribute a **property_process** that is referenced through the **process** attribute by a **process_-property_association** with a description of 'fixture assembly element shape' that references through the **property_or_shape** attribute a **shape_aspect**.

## 5.2.3.2.34 fixture_assembly_relationship

A **fixture_assembly_relationship** is an **action_resource_relationship** which relates a fixture assembly element to a fixture assembly.

EXPRESS specification:

```
*)
ENTITY fixture_assembly_relationship
   SUBTYPE OF (action_resource_relationship);
WHERE
 wr1: 'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY'
    IN TYPEOF(SELF.relating_resource);
 wr2: (SIZEOF([
   'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
   'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT'] *
    TYPEOF(SELF.related_resource)) =1) ;

END_ENTITY;
(*
```

Formal proposition:

**WR1**: The **relating_resource** is a **fixture_assembly**.

**WR2**: The **related_resource** is a **fixture_assembly** or **fixture_assembly_element**.

## 5.2.3.2.35 fixture_contract

A **fixture_contract** is a type of **contract** that defines documentation for fixtures.

EXPRESS specification:

```
*)
ENTITY fixture_contract
   SUBTYPE OF (contract);
WHERE

   (* fixture_contract.design_order *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')|
   (adr.assigned_action_request.description = 'design order'))) = 1;
```

```
   (* fixture_contract.fabrication_order *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS') |
   (adr.assigned_action_request.description = 'fabrication order'))) = 1;

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')|
   (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
   IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items |
   (('PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION' IN TYPEOF(it))
   ) )) <= 1 )) )) ) = 0;


END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall exactly one references from a **applied_action_request_assignment** of kind **actio_-request_assignment** through the **items** attribute that references a **versioned_action_request** with a **name** of 'design order' through the **assigned_action_request**.

**WR2:** There shall exactly one references from a **applied_action_request_assignment** of kind **actio_-request_assignment** through the **items** attribute that references a **versioned_action_request** with a **name** of 'fabrication order' through the **assigned_action_request**.

**WR3:** There shall be referenced by exactly one **applied_contract_assignment** of kind **contract_-assignment** through the **assigned_contract** that references zero or one **in_facility_location** in the list of **items**.

## 5.2.3.2.36 fixture_machine_mounting

A **fixture_machine_mounting** is a **fixture_setup** which is the technique of locating the product on the machine.

EXPRESS specification:

```
*)
ENTITY fixture_machine_mounting
   SUBTYPE OF (fixture_setup);

END_ENTITY;
(*
```

## 5.2.3.2.37 fixture_machine_unmounting

A **fixture_machine_unmounting** is a **fixture_setup** which is the technique of locating the product on the machine.

 

EXPRESS specification:

```
*)
ENTITY fixture_machine_unmounting
   SUBTYPE OF (fixture_setup);

END_ENTITY;
(*
```

## 5.2.3.2.38 fixture_pallet_mounting

A **fixture_pallet_mounting** is a **fixture_setup** which is the technique of locating the product on the machine pallet.

EXPRESS specification:

```
*)
 ENTITY fixture_pallet_mounting
  SUBTYPE OF (fixture_setup);

  WHERE
   wr1: (NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 0)))
      = 0));
 END_ENTITY; -- fixture_pallet_mounting
(*
```

Formal propositions:

**WR1:** Then there shall be exactly zero reference by an **action_resource_requirement** of type **requirement_for_action_resource** through the **operations** that references a **machine** through **resources**.

## 5.2.3.2.39 fixture_pallet_unmounting

A **fixture_pallet_unmounting** is a **fixture_setup** which is the technique of locating the product on the machine pallet.

EXPRESS specification:

```
*)
 ENTITY fixture_pallet_unmounting
  SUBTYPE OF (fixture_setup);

  WHERE
   wr1: (NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'  IN  TYPEOF(arr)))
|
      (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 0)))
      = 0));
 END_ENTITY; -- fixture_pallet_unmounting
(*
```

Formal propositions:

**WR1:** Then there shall be exactly zero reference by an **action_resource_requirement** of type **requirement_for_action_resource** through the **operations** that references a **machine** through **resources**.

## 5.2.3.2.40 fixture_setup

A **fixture_setup** is an **action_method** which places or locates the fixture assembly on the machine.

EXPRESS specification:

```
*)
ENTITY fixture_setup
   SUBTYPE OF (process_plan_activity);
WHERE

   wr1:   SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
        'PROCESS_PLANNING_SCHEMA.MOUNTING_POSITION' IN TYPEOF(ap)) ))
        <= 1;
   wr2:  ((NOT type_check_function(
       SELF ,['PROCESS_PLANNING_SCHEMA.FIXTURE_PALLET_MOUNTING',
        'PROCESS_PLANNING_SCHEMA.FIXTURE_PALLET_UNMOUNTING'],1)) OR
      (NOT(SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 1)))
      = 0)));

   wr3: (NOT(SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.PALLET' IN TYPEOF(it)) )) >= 0)))
      = 0));
```

 

```
   wr4: (NOT(SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (SIZEOF(QUERY ( it <* am.resources |
    ('PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(it)) )) >= 1)))
      = 0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be zero or one reference by an **action_property** through **definition** that is of type **mounting_position**.

**WR2:** If this is not a **fixture_pallet_mounting** or fixture_pallet**_unmounting** then there shall be exactly one reference by an **action_resource_requirement** of type **requirement_for_action_resource** through the **operations** that references a **machine** through **resources**.

**WR3:** There shall be exactly one reference by an **action_resource_requirement** of type **requirement_-for_action_resource** through the **operations** that references one **pallet** through **resources**.

**WR4:** There shall be one or more reference by an **action_resource_requirement** of type **requirement_-for_action_resource** through the **operations** that references one or more **fixture_assembly**s through **resources**.

## 5.2.3.2.41 generic_manufacturing_resource

A **generic_manufacturing_resource** is a type of **action_resource** that defines resources for process planning activities.

EXPRESS specification:

```
*)
ENTITY generic_manufacturing_resource
 SUPERTYPE OF (ONEOF (resource_with_representation, resource_with_material))
   SUBTYPE OF (action_resource);
WHERE


 wr1: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( rpr <* apr.representation.items |
   ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
   TYPEOF(rpr)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(rpr\measure_with_unit.value_component)) AND (rpr.name
   = 'resource quantity')
   ))>=1)
   )) = 0)))) = 0));


 wr2: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
```

```
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( rpr <* apr.representation.items |
   ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
   TYPEOF(rpr)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(rpr\measure_with_unit.value_component)) AND (rpr.name
   = 'resource quantity') AND
    (rpr\measure_with_unit.unit_component.name = 'each')
   )) = 1))) = 0)))) = 0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **measure_representation_item** and **count_measure** with a **name** of 'resource quantity'.

**WR2:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **measure_representation_item** and **unit_component** of **count_measure** with a **name** of 'each'.

## 5.2.3.2.42 illustration

A **illustration** is a type of **document** that is a pictorial presentation of a part.

EXPRESS specification:

```
*)
ENTITY illustration
   SUBTYPE OF (document);
WHERE

  (* illustration to activity *)

 wr1: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
   ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
   IN TYPEOF(dr))) |
   (NOT(SIZEOF(QUERY ( d <* adr.items |
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d))
   )) >= 1)))) =0;

 (* illustration to view_reference *)

 wr2: SIZEOF(QUERY ( aduc <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS')|
    ('PROCESS_PLANNING_SCHEMA.VIEW_REFERENCE'
    IN TYPEOF
    (aduc\document_usage_constraint_assignment.assigned_document_usage))
    )) >= 0;

END_ENTITY;
(*
```

 

Formal propositions:

**WR1:** The **illustration** shall be the **assigned_document** in exactly one **applied_document_reference** that has in the list of **items** one or more **process_planning_activity**.

**WR2:** A **illustration** shall be referenced by one or more **document_usage_constraint** of kind **view_reference**.

## 5.2.3.2.43 in_facility_location

The **in_facility_location** is a type of **resource_property** that defines the position of a **work_cell**.

EXPRESS specification:

```
*)
ENTITY in_facility_location
   SUBTYPE OF (resource_property);
WHERE

 (* in_facility_location.building_or_area *)

 wr1: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* ap.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'building or area'))))=1)))=0));

 (* in_facility_location.location_code *)

 wr2: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* ap.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
   TYPEOF(it)) AND (it.name = 'location code'))))=1)))=0));

 (* in_facility_location.sublocation *)

 wr3: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* ap.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'sublocation')))) = 1)))=0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation_item** used for the representation of an **in_facility_location** shall be of type **descriptive_representation_item** with a name of 'building or area'.

**WR2:** Exactly one **representation_item** used for the representation of an **in_facility_location** shall be of type **descriptive_representation_item** with a name of 'location code'.

**WR3:** Exactly one **representation_item** used for the representation of an **in_facility_location** shall be of type **descriptive_representation_item** with a name of 'sublocation'

## 5.2.3.2.44 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.

EXPRESS specification:

```
*)
ENTITY known_source
   SUBTYPE OF (external_source,pre_defined_item);
END_ENTITY;
(*
```

## 5.2.3.2.45 library_class_version_assignment

An **library_class_version_assignment** is a type of **applied_external_identification_assignment** that specifies a type of external identification.

EXPRESS specification:

```
*)
ENTITY library_class_version_assignment
   SUBTYPE OF (applied_external_identification_assignment);
END_ENTITY;
(*
```

## 5.2.3.2.46 library_property_version_assignment

An **library_property_version_assignment** is a type of **applied_external_identification_assignment** that specifies a type of external identification.

EXPRESS specification:

```
*)
ENTITY library_property_version_assignment
   SUBTYPE OF (applied_external_identification_assignment);
END_ENTITY;
(*
```

## 5.2.3.2.47 machine

A **machine** is a type of **action_resource** that defines machines for process planning activities.

EXPRESS specification:

```
*)
ENTITY machine
   SUBTYPE OF (action_resource);
WHERE
```

 

```
 (* machine.company_name *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.ORGANIZATION'
   IN TYPEOF(adr.assigned_organization)) AND
   (adr.assigned_organization.description = 'machine company name')
   ))) = 1;

 (* machine to controller *)

 wr2: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')|
   ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
   IN TYPEOF(arr))) |
   ('PROCESS_PLANNING_SCHEMA.CONTROLLER' IN TYPEOF(ar2.relating_resource)
   ))) <= 1;


 (* machine to workstation *)

 wr3: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')|
   ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
   IN TYPEOF(arr))) |
   ('PROCESS_PLANNING_SCHEMA.WORKSTATION' IN TYPEOF(ar2.relating_resource)
   ))) = 1;

 (* machine to pallet *)


 wr4: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')|
   ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
   IN TYPEOF(arr))) |
   ('PROCESS_PLANNING_SCHEMA.PALLET' IN TYPEOF(ar2.relating_resource)
   ))) = 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall exactly one references from a **applied_organization_assignment** through the **items** attribute that references a **organization** with a **description** of 'machine company name' of through the **assigned_organization**.

**WR2:** There shall be one or more references from a **action_resource_relationship** of type **tool_assembly_relationship** through **related_resource** that references a **controller** through **relating_resource**.

**WR3:** There shall be one or more references from a **action_resource_relationship** of type **tool_assembly_relationship** through **related_resource** that references a **workstation** through **relating_resource**.

**WR4:** There shall be one or more references from a **action_resource_relationship** of type **tool_assembly_relationship** through **related_resource** that references a **pallet** through **relating_resource**.

## 5.2.3.2.48 machine_element_relationship

A machine_element_relationship is an **action_resource_relationship** which relates a machine to a machine components.

EXPRESS specification:

```
*)
ENTITY machine_element_relationship
   SUBTYPE OF (action_resource_relationship);
WHERE
 wr1: 'PROCESS_PLANNING_SCHEMA.MACHINE'
    IN TYPEOF(SELF.related_resource);
 wr2: (SIZEOF([
   'PROCESS_PLANNING_SCHEMA.CONTROLLER',
   'PROCESS_PLANNING_SCHEMA.PALLET',
   'PROCESS_PLANNING_SCHEMA.WORKSTATION'] *
    TYPEOF(SELF.relating_resource)) =1) ;
END_ENTITY;
(*
```

Formal proposition:

**WR1**: The **relating_resource** is a **fixture_assembly**.

**WR2**: The **related_resource** is a **controller**, **pallet** or **workstation**.

## 5.2.3.2.49 machine_setup

A **machine_setup** is an **action_method** which places or locates the fixture assembly and the part on the machine.

EXPRESS specification:

```
*)
ENTITY machine_setup
   SUBTYPE OF (process_plan_activity);
WHERE
  (* machine_setup to machine *)

 wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS') |
   ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
   IN TYPEOF(arr))) | (NOT (SIZEOF(QUERY ( it <* am.resources |
   ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it))
   )) >= 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 **machine** through **resources**.

## 5.2.3.2.50 machine_usage

A **machine_usage** is a type of **requirement_for_action_resource** that defines requirements for **machine** and **workstation**.

EXPRESS specification:

```
*)
ENTITY machine_usage
   SUBTYPE OF (requirement_for_action_resource);
WHERE
   (* machine_usage to machine *)


 wr1: SIZEOF(QUERY( w <* SELF.resources |
    ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(W))))=1;

 (* machine_usage to specification *)

 wr2: SIZEOF(QUERY (adr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
    ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN TYPEOF
    (adr.assigned_document))))<=1;

 (* machine_usage to machine_parameters*)

 wr3: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name='maximum spindle speed')
   ) ))<=1)) ))=0))))=0));

 wr4: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name='maximum feedrate')
```

```
  ) ))<=1)) ))=0))))=0));

wr5: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
   IN TYPEOF(pdr.representation))) |
   ((impl_rep.representation.name = 'machine usage parameter')AND
   (SIZEOF(QUERY ( it <* impl_rep.representation.items |
  ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
  TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
  TYPEOF(it\measure_with_unit.value_component)) AND (it.name
  = 'number of control axis')
  ))<=1)) ))=0))))=0));

 wr6: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
      ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
      IN TYPEOF(pdr.representation))) |
      ((impl_rep.representation.name = 'machine usage parameter')AND
      (SIZEOF(QUERY ( it <* impl_rep.representation.items |
     ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND  (it.name  IN
     ['x-axis table size','y-axis table size'])
     ) ))<=2))  ))=0))))=0));

wr7: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
   IN TYPEOF(pdr.representation))) |
   ((impl_rep.representation.name = 'machine usage parameter')AND
   (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.description IN ['true','false','unknown'])
   AND (it.name='table indexing function')
  )))<=1)) ))=0))))=0));


wr8: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
   IN TYPEOF(pdr.representation))) |
   ((impl_rep.representation.name = 'machine usage parameter')AND
   (SIZEOF(QUERY ( it <* impl_rep.representation.items |
  ((SIZEOF([
  'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
  'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
  TYPEOF(it)) = 2) AND (it.name='positioning accuracy')
  ) ))<=1)) ))=0))))=0));

wr9: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
```

 

```
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name='axis range of motion ')
   ) ))>=0)) ))=0))))=0));

 wr10: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation)) ) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
    (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
    IN TYPEOF(it)) AND (it.name='axis range of motion description')
    )))>=0))  ))=0))))=0));

 wr11: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
   TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(it\measure_with_unit.value_component)) AND (it.name
   = 'number of simultaneous axis')
   )) <=1)) ))=0))))=0));

 wr12: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
    ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
    IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'machine usage parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name='spindle power')
   ) ))<=1)) ))=0))))=0));


END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be one **resources** of type **machine**.

**WR2:** The **machine_usage** 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.

**WR3:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **ratio_representation_item** with a **name** of 'maximum spindle speed'.

**WR4:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **length_representation_item** with a **name** of 'maximum feedrate'.

**WR5:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **count_measure** with a **name** of 'number of control axis'.

**WR6:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **length_representation_item** with a **name** of 'x-axis table size' or 'y-axis table size'.

**WR7:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **descriptive_-representation_item** with a **name** of 'table indexing function' and a **description** of either 'true', 'false', or 'unknown'.

**WR8:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **length_representation_item** with a **name** of 'positioning accuracy'.

**WR9:** There shall be two or more reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **length_representation_item** with a **name** of 'axis range of motion '.

**WR10:** There shall be two or more reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **descriptive_-representation_item** with a **name** of 'axis range of motion description'.

 

**WR11:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **count_measure** with a **name** of 'number of simultaneous axis'.

**WR12:** There shall be exactly one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **ratio_representation_item** with a **name** of 'spindle power'.

## 5.2.3.2.51 machining_tolerance

An **machining_tolerance** is an **action_property** which is the description of the deviation allowed for a cut segment.

EXPRESS specification:

```
*)
ENTITY machining_tolerance
 SUBTYPE OF (action_property);
WHERE

 wr1: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name = 'inside tolerance'))))=1)))=0));

 wr2: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name = 'outside tolerance'))))=1)))=0));

 wr3: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name = 'total tolerance'))))<=1)))=0));
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one **representation_item** with the name of: ‘inside tolerance’.

**WR2:** There shall be exactly one **representation_item** with the name of: ‘outside tolerance’.

**WR3:** There shall be exactly one **representation_item** with the name of: ‘total tolerance’.

## 5.2.3.2.52 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**.

EXPRESS specification:

```
*)
 ENTITY manufacturing_activity
  SUPERTYPE OF (ONEOF (ancillary_activity, part_routing,
  part_unmounting, process_activity, validation, part_mounting ))
  SUBTYPE OF (process_plan_activity);
  WHERE
 (* manufacturing_activity to controller_program *)

 wr1: (NOT ((
   'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS')
   IN TYPEOF(SELF))) OR
   (SIZEOF(QUERY ( adr <*
   SELF\action_method_with_associated_documents.documents |
   ('PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM' IN TYPEOF(adr))
   )) <= 1);
   wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')|
      (('PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
      IN TYPEOF(sar)) AND
      (NOT(SIZEOF(QUERY ( edi <* sar.items |
      ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_SCHEMA'
      IN TYPEOF(edi)) )) <= 1)) )) )=0;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be zero or one **controller** referenced through **documents**.

**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 **externally_defined_schema**

## 5.2.3.2.53 manufacturing_activity_relationship

The **manufacturing_activity_relationship** is a type of **sequential_method** that relates a **manufacturing_process** to either a **manufacturing_activity**.

EXPRESS specification:

```
*)
ENTITY manufacturing_activity_relationship
   SUBTYPE OF (sequential_method);
WHERE

 wr1: ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
   IN TYPEOF(SELF.relating_method));
```

 

```
 wr2:   ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY'
   IN TYPEOF(SELF.related_method));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **related_method** attribute shall be **manufacturing_process**.

**WR2:** The **relating_method** attribute shall be **manufacturing_activity**.

## 5.2.3.2.54 mounting_position

The **mounting_position** is a type of **action_property** that defines the placement and orientation of a part or fixture.

EXPRESS specification:

```
*)
ENTITY mounting_position
  SUBTYPE OF (action_property);
WHERE

 wr1: (NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND
   (it.name = 'orientation'))))=1)))=0));

 wr2: (NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(it)) AND
   (it.name = 'location origin'))))=1)))=0));

 wr3: (NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.PLANE' IN TYPEOF(it)) AND
   (it.name = 'reference plane'))))=1)))=0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that has **placement** with a **name** of ' location origin' in the list of **items** with a **cartesian_point**.

**WR2:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that has **placement** with a **name** of 'orientation' in the list of **items**.

**WR3:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that has **plane** with a **name** of 'reference plane in the list of **items**.

## 5.2.3.2.55 pallet

A **pallet** is a type of **action_resource** that is a working surface associated with a machine to place a piece part being manufactured

EXPRESS specification:

```
*)
ENTITY pallet
  SUBTYPE OF (action_resource);

END_ENTITY;
(*
```

## 5.2.3.2.56 pallet_machine_mounting

A **pallet_machine_mounting** is a **part_mounting** which is the technique of locating the pallet on the machine.

EXPRESS specification:

```
*)
ENTITY pallet_machine_mounting
   SUBTYPE OF (part_mounting);
WHERE
 wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
   'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) ))
   = 1;
 wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
   'PROCESS_PLANNING_SCHEMA.PALLET' IN TYPEOF(ap)) ))
   = 1;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **machine** through **usage**.

**WR2:** There shall be exactly one reference by a **pallet** through **usage**.

 

## 5.2.3.2.57 ngon_shape_representation

An **ngon_shape_representation** specifies representation of a shape that is a volume defined as a ngon area of a defined length. The enclosed area is defined by three or more straight sides.

EXPRESS specification:

```
*)
 ENTITY ngon_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
 wr1: SIZEOF(SELF.items) = 5;
 wr2: SIZEOF(QUERY ( it <* SELF.items | ((
   'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
   name = 'orientation')))) = 1;
 wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name = 'length')))) = 1;
 wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name = 'corner radius')))) = 1;
 wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
   TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
   'diameter across flats'])))) = 1;
 wr6: SIZEOF(QUERY ( it <* SELF.items | (((
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
   TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(it\measure_with_unit.value_component))) AND (it.name
   = 'number of sides')))) = 1;
 END_ENTITY; -- ngon_shape_representation
(*
```

Formal propositions:

**WR1:** The **ngon_shape_representation** shall contain exactly five **representation_item**s in its set of **items**.

**WR2:** One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **placement** with a **name** of 'orientation'.

**WR3:** One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'length'.

**WR4:** One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'corner radius'.

**WR5:** One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'circumscribed diameter' or 'diameter across flats'.

**WR6:** One of the **representation_item**s used for the implicit representation of a **ngon_shape_-representation** shall be of type **measure_representation_item** with a **value_component** of type **count_-measure** and a **name** of 'number of sides'.

Informal propositions:

**IP1:** The location of the **ngon_shape_representation** shall be defined at the center of the enclosed area.

**IP2**: The **ngon_shape_representation** shall be defined with the enclosed area in the X-Y plane with one of the sides of the ngon parallel to the X direction intersecting the negative Y axis. The length is along the Z direction.

## 5.2.3.2.58 non_machining_process

The **non_machining_process** is a type of **manufacturing_process** that defines **manufacturing_activity**s for non machining processes.

EXPRESS specification:

```
*)
ENTITY non_machining_process

   SUBTYPE OF (manufacturing_process);


END_ENTITY;
(*
```

## 5.2.3.2.59 part_routing

A **part_routing** is an **action_method** which is the method of relocating the product from one discrete location to another.

EXPRESS specification:

```
*)
 ENTITY part_routing
  SUBTYPE OF (manufacturing_activity);


 END_ENTITY; -- part_routing
(*
```

## 5.2.3.2.60 part_contract

A **part_contract** is a type of **contract** that defines documentation to manufacture a part.

 

EXPRESS specification:

```
*)
ENTITY part_contract
   SUBTYPE OF (contract);
WHERE

 (* contract to company *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.ORGANIZATION'
   IN TYPEOF(adr.assigned_organization)) AND
   (adr.assigned_organization.description='binding aggrement'))))=1;

 (* contract to part_version *)

 wr2: SIZEOF(QUERY ( ca <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')|
   (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
   IN TYPEOF(ca)) AND
   (NOT (SIZEOF(QUERY ( pdf <* ca.items |
   (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_FORMATION'
   IN TYPEOF(pdf)) AND
   (NOT (SIZEOF(QUERY ( pd <* USEDIN(pdf,
   'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION.FORMATION') |
   (pd.frame_of_reference.name = 'part definition'))) = 1)))))
   = 1))))) = 0;

 (* contract to document_assignment *)

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')|
   (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
   IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items |
   (('PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE' IN TYPEOF(it))
   AND (it.description = 'contract document')))) <= 1
   ))))) = 0;

 (* contract.project_number *)

 wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.ORGANIZATION'
   IN TYPEOF(adr.assigned_organization)) AND
   (SIZEOF(QUERY ( op <* USEDIN(adr.assigned_organization,
'PROCESS_PLANNING_SCHEMA.ORGANIZATIONAL_PROJECT.RESPONSIBLE_ORGANIZATIONS')
|
   (op.description = 'project number'))) = 1)))) = 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be referenced by exactly one applied_organization_assignment of kind **organization_-assignment** through the **assigned_organization** that references exactly one **organization** with a **name** of 'binding agreement' in the list of **items**.

**WR2:** There shall be referenced by exactly one **applied_contract_assignment** of kind **contract_-assignment** through the **assigned_contract** that references exactly one **product_definition_formation** with a **product_definition_context** with a **name** of 'part definition'

**WR3:** There shall be referenced by exactly one **applied_contract_assignment** of kind **contract_-assignment** through the **assigned_contract** that references zero or one **document_file** with a **description** of 'contract document' in the list of **items**.

**WR4:** There shall be referenced by exactly one **applied_organization_assignment** of kind **organization_assignment** through the **assigned_organization** that references exactly one **organization** that is referenced by an **organizational_project** through **responsible_organization** with a **description** of 'project number' in the list of **items**.

## 5.2.3.2.61 part_fixture_mounting

A **part_fixture_mounting** is an **action_method** which is the technique of locating the product in a holding device.

EXPRESS specification:

```
*)
ENTITY part_fixture_mounting
   SUBTYPE OF (part_mounting);
WHERE
 wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
   'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(ap)) ))
   >= 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **fixture_assembly** through **usage**.

## 5.2.3.2.62 part_mounting

A **part_mounting** is a **manufacturing_activity** that is a method of loading a part on a machine, fixture, or pallet.

EXPRESS specification:

```
*)
ENTITY part_mounting
   SUPERTYPE OF (ONEOF (part_fixture_mounting,
             part_machine_mounting,
             pallet_machine_mounting ))
   SUBTYPE OF (manufacturing_activity);
WHERE
```

 

```
   wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
      (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
      (SIZEOF(QUERY ( it <* apr.representation.items |
      (('PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(it)) AND
      (it.name = 'part location origin')) )) = 1) )) = 1) )) = 1;

   wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.MOUNTING_POSITION' IN TYPEOF(ap)) ))
      <= 1;
   wr3: (SIZEOF(QUERY ( cm <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
      (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
      'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
      (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF
      (ppa.property_or_shape)) AND
      (ppa.description = 'part mounting part shape')))) <= 1))
      )) = 0);
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that has **placement** with a **name** of 'fixture to machine location' in the list of **items** with a **cartesian_point**with a **name** of 'part location origin'.

**WR2:** There shall be exactly one reference by an **action_property** through **definition** that is of type **mounting_position**.

**WR3:** There shall be exactly one reference by an **action** through **chosen_method** that is referenced by a **process_property_association** through **process** with a **description** of 'part mounting part shape' that references **product_definition_shape** through **property_or_shape**.

## 5.2.3.2.63 part_unmounting

A **part_unmounting** is an **manufacturing_activity** which is the technique of removing the product in a holding device.

EXPRESS specification:

```
*)
ENTITY part_unmounting
   SUBTYPE OF (manufacturing_activity);
WHERE
 wr1: (SELF.name IN ['pallet machine unmounting',
           'part fixture unmounting',
           'part machine unmounting']);
END_ENTITY;
(*
```

Formal propositions:

**WR1**: The **name** attribute shall be one the following 'pallet machine unmounting', 'part fixture unmounting', or 'part machine unmounting'.

## 5.2.3.2.64 part_machine_mounting

A **part_machine_mounting** is an **action_method** which is the technique of locating the product on the machine.

EXPRESS specification:

```
*)
ENTITY part_machine_mounting
   SUBTYPE OF (part_mounting);
WHERE

   wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
        'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) ))
        = 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 **machine** through **resources**.

## 5.2.3.2.65 process_activity

The **process_activity** is a type of **manufacturing_activity** is a type of **process_plan_activity**, that is a type of **action_method** which is the definition of a machining process planning **Activity**.

EXPRESS specification:

```
*)
ENTITY process_activity
   SUBTYPE OF (manufacturing_activity);
WHERE
 (* process_activity to property_parameter *)

   wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')|
      (('PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
        IN TYPEOF(sar)) AND (NOT (SIZEOF(QUERY ( edi <* sar.items |
        ('PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION' IN
        TYPEOF(edi)) )) >= 0))) )) = 0;
 (* process_activity to shape_aspect *)

 wr2: SIZEOF(QUERY ( cm <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
   (('PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS'
   IN TYPEOF(cm)) AND (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
```

 

```
    'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
    ((ppa.description = 'process activity part shape') AND
    ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
    IN TYPEOF(ppa.property_or_shape))))) <=1)))))=0;

  (* process_activity to machine_setup *)

  wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
    ('PROCESS_PLANNING_SCHEMA.DEFINING_ACTION_METHOD_RELATIONSHIP'
    IN TYPEOF(sar)) AND
    ('PROCESS_PLANNING_SCHEMA.MACHINE_SETUP'
    IN TYPEOF(sar.related_method))))<=1;

  (* process_activity to fixture_assembly *)

  wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
    'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(ap))))
    <= 1;

  (* process_activity to tool_assembly *)

  wr5: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
    'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(ap))
    )) >= 1;


  (* process_activity to machining_tolerance *)

  wr6: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
    'PROCESS_PLANNING_SCHEMA.MACHINING_TOLERANCE' IN TYPEOF(ap))))
    <= 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 **property_definition**

**WR2:** There shall be exactly one reference by an **action** through **chosen_method** that is referenced by a **process_property_association** through **process** with a **description** of 'process activity part shape' that references **product_definition_shape** through **property_or_shape**, and this **product_definition_shape** is referenced by a **shape_aspect** through **of_shape**.

**WR3:** There shall be exactly one references from an **action_method_relationship** through **relating_method** that references a **machine_setup** through **related_method**.

**WR4:** There shall be exactly one reference by a **fixture_assembly** through **usage**.

**WR5:** There shall be exactly one reference by a **tool_assembly** through **usage**.

**WR6:** There shall be exactly one reference by an **action_property** through **definition** that is of type **machining_tolerance**.

## 5.2.3.2.66 process_plan_activity

The **process_plan_activity** is a type of **action_method_with_associated_documents** which defines **action_method**s required to define a process plan type of **action**.

EXPRESS specification:

```
*)
 ENTITY process_plan_activity
  SUPERTYPE OF (ONEOF (ancillary_setup,fixture_setup,machine_setup,
    manufacturing_activity,tool_setup))
  SUBTYPE OF (action_method);
  WHERE

 (* process_plan_activity.frequency *)

 wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') |
   (ap.name = 'frequency'))) <= 1;

 (* process_plan_activity.organization_id *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS') |
   (('PROCESS_PLANNING_SCHEMA.ORGANIZATION'
   IN TYPEOF(adr.assigned_organization)) AND
   (adr.role.name = 'activity organization id'))
   )) = 1;

 (* process_plan_activity to special_instruction *)

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   NOT (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'special instruction'))
   )) >= 0))) = 0)))) = 0;

 (* process_plan_activity to supplemental_document *)

 wr4: (NOT ((
   'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS')
   IN TYPEOF(SELF)))
   OR (SIZEOF(QUERY ( adr <*
   SELF\action_method_with_associated_documents.documents |
```

 

```
  (('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION'
  IN TYPEOF(adr)) AND
  (adr.kind.product_data_type = 'supplemental documents')))) >= 0);

(* process_plan_activity.graphics_representation *)


wr5: (NOT ((
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS')
  IN TYPEOF(SELF)))
  OR (SIZEOF(QUERY ( adr <*
  SELF\action_method_with_associated_documents.documents |
  ('PROCESS_PLANNING_SCHEMA.ILLUSTRATION' IN TYPEOF(adr)))) >=0);


wr6: (NOT ((
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS')
  IN TYPEOF(SELF)))
  OR (SIZEOF(QUERY ( adr <*
  SELF\action_method_with_associated_documents.documents |
  ('PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(adr)))) >=0);


wr7: (NOT ((
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS')
  IN TYPEOF(SELF))) OR
  (SIZEOF(QUERY(edi <*
  SELF\action_method_with_associated_documents.documents |
  ('PROCESS_PLANNING_SCHEMA.VIEW_REFERENCE' IN TYPEOF(edi)
  )))>=0);

(* process_plan_activity to performance_rate *)

wr8: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
  (SIZEOF(['PROCESS_PLANNING_SCHEMA.PRODUCTION_RATE',
  'PROCESS_PLANNING_SCHEMA.ALLOWED_TIME'] * TYPEOF(ap))=1)
  )) >=1;


(* process_plan_activity to generic_manufacturing_resource *)

wr9: SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') |
  ('PROCESS_PLANNING_SCHEMA.GENERIC_MANUFACTURING_RESOURCE'
  IN TYPEOF(ap)))) >= 0;

(* process_plan_activity to material *)

wr10: SIZEOF(QUERY ( cm <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
  (('PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS'
  IN TYPEOF(cm)) AND (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
  'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
  ((ppa.description = 'process plan activity material') AND
  ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION'
  IN TYPEOF(ppa.property_or_shape.definition))))) <=1)))))=0;

wr11: (NOT (('PROCESS_PLANNING_SCHEMA.' +
   'ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS') IN TYPEOF(SELF)))
  OR (SIZEOF(SELF\action_method_with_associated_documents.documents) >= 0);
```

```
 wr12: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   NOT (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(it\measure_with_unit.value_component))AND
   (it.name = 'activity number'))
   )) >= 0))) = 0)))) = 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **action_resource** through **usage** with a name of 'frequency'.

**WR2:** There shall exactly one references from a **applied_organization_assignment** through the **items** attribute that references a **organization** with an **organization_role** with **name** of 'activity organization id' of through the **assigned_organization**.

**WR3:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **descriptive_representation_item** with a **name** of 'special instruction'.

**WR4:** There shall be zero or one **document** with a **name** of 'supplemental documents' referenced through **documents**.

**WR5:** There shall be zero or one **design_reference** referenced through **documents**.

**WR6:** There shall be zero or one **illustration** referenced through **documents**.

**WR7:** There shall be zero or one reference through the **documents** attribute of a **action_method_with_-associated_documents** that is referenced through the **source** attribute of a **view_reference**.

**WR8:** There shall be zero or more references by a **production_rate** of kind **action_property** through **definition**.

**WR9:** There shall be zero or more references by a **action_resource** of kind **generic_manufacturing_-resource** through **usage**.

**WR10:** There shall zero or one reference from a **action** through **chosen_method** 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**.

**WR11:** There shall be zero or more **documents** referenced by **process_plan_activity** through the attribute of **documents** for **action_method_with_associated_documents**.

 

**WR12:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **measure_representation_item** and **count_measure** with a **name** of 'activity number'.

## 5.2.3.2.67 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**.

EXPRESS specification:

```
*)
ENTITY process_plan_security
  SUBTYPE OF (security_classification);
WHERE

 (* process_plan_security.classification_date *)

 wr1: SIZEOF(QUERY ( da <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
   ((('PROCESS_PLANNING_SCHEMA.DATE_ASSIGNMENT'
   IN TYPEOF(da)) AND
   ('PROCESS_PLANNING_SCHEMA.CALENDAR_DATE'
   IN TYPEOF(da.assigned_date))) AND
   (da.role.name = 'classification date'))))
   = 1;

 (* process_plan_security.declassification_date *)

 wr2: SIZEOF(QUERY ( da <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') |
   ((('PROCESS_PLANNING_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da))
   AND ('PROCESS_PLANNING_SCHEMA.CALENDAR_DATE'
   IN TYPEOF(da.assigned_date))) AND
   (da.role.name = 'declassification date'))))
   = 1;

 (* process_plan_security to activity *)

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.SECURITY_CLASSIFICATION_ASSIGNMENT.'+
   'ASSIGNED_SECURITY_CLASSIFICATION') |
   (('PROCESS_PLANNING_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
   IN TYPEOF(ap)) AND
   (NOT (SIZEOF(QUERY ( it <* ap.items |
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(it))))
   = 1))))) = 0;
```

```
 (* process_plan_security to process_plan_version *)

 wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
   'SECURITY_CLASSIFICATION_ASSIGNMENT.ASSIGNED_SECURITY_CLASSIFICATION')
    | (('PROCESS_PLANNING_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
   IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (
   'PROCESS_PLANNING_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_assignement** 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_assignement** that references a **process_plan_version** through **items**.

## 5.2.3.2.68 process_plan_specification

The **process_plan_specification** is a type of **document** that defines the subclass and revision data for process plan specifications.

EXPRESS specification:

```
*)

 ENTITY process_plan_specification
   SUBTYPE OF (document);
 WHERE
 wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')|
   (duc.subject_element = 'revision')))=1;
 wr2: SIZEOF(QUERY ( duc <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')|
   (duc.subject_element = 'subclass')))=1;
 END_ENTITY;
(*
```

 

Formal propositions:

**WR1:** The **tool_assembly_element** shall be in the set of **items** of exactly one **applied_document_usage_-constraint_assignment** with a **subject_element** of 'revision' that defines the **document_file** containing the controller specification.

**WR2:** The **tool_assembly_element** shall be in the set of **items** of exactly one **applied_document_usage_-constraint_assignment** with a **subject_element** of 'subclass' that defines the **document_file** containing the controller specification.

## 5.2.3.2.69 process_plan_version

An **process_plan_version** relates an action to a **product_property_process** and a **product_definition_-process** to a particular application context.

EXPRESS specification:

```
*)
ENTITY process_plan_version
   SUBTYPE OF (product_definition_process);

WHERE

 (* process_plan_version.manufacturing_GT_code *)

 wr1 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
    (ap.description = 'manufacturing gt code'))) <= 1;

 (* process_plan_version.quantity_of_parts*)

 wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   NOT (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
   TYPEOF(it\measure_with_unit.value_component))AND
   (it.name = 'quantity of parts'))
   )) >= 0))) = 0)))) = 0;

 (* process_plan_version to special_instruction *)

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   NOT (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'special instruction'))
   )) >= 0))) = 0)))) = 0;

 (* process_plan_version to property_parameter *)

 wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') |
```

```
   (NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   NOT (SIZEOF(QUERY ( it <* apr.representation.items |
   (it.name = 'auxilary header')
   )) >= 0))) = 0)))) = 0;

 (* process_plan_version to range_of_parts *)

 wr5 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
    'PROCESS_PLANNING_SCHEMA.RANGE_OF_PARTS' IN TYPEOF(ap))))
    <= 1;

 (* process_plan_version to material*)

 wr6 : SIZEOF(QUERY ( ppp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.PROCESS') |
    (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION'
    IN TYPEOF(ppp.defined_product)) AND
    (ppp.defined_product.name='material'))
   )) <= 1;

 (* process_plan_version to manufacturing_process *)

 wr7: (SIZEOF(['PROCESS_PLANNING_SCHEMA.NON_MACHINING_PROCESS',
    'PROCESS_PLANNING_SCHEMA.MACHINING_PROCESS']
    * TYPEOF(SELF.chosen_method))=1);

   wr8: SIZEOF([
        'PROCESS_PLANNING_SCHEMA.NON_MACHINING_PROCESS',
'PROCESS_PLANNING_SCHEMA.MACHINING_PROCESS'] * TYPEOF(SELF.chosen_method))=1;

 (* process_plan_version to feature_dependency *)

 wr9: SIZEOF(QUERY ( ap <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION') |
    (('PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_ASSIGNMENT'
    IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items |
    ('PROCESS_PLANNING_SCHEMA.FEATURE_DEPENDENCY' IN TYPEOF(it))))
    <= 1))))) = 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by a **action_property** through **definiton** with a description of 'manufacturing gt code'.

**WR2:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **descriptive_representation_item** with a **name** of 'quantity of parts'.

**WR3:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **descriptive_representation_item** with a **name** of 'special instruction'.

 

**WR4:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **representation_item** with a **name** of 'auxilary header'.

**WR5:** There shall be zero or more references by a **range_of_parts** of kind **action_property** through **definition**.

**WR6:** There shall be zero or more references by a **process_product_association** through **process** that references a **product_definition** through **defined_product**.

**WR7**: There shall be exactly one **machining_process** or **non_machining_process** referenced through the **chosen_method** attribute.

**WR8:** There shall be exactly one **manufacturing_process** of type **machining_process** or **non_-machining_process** referenced through the **chosen_method** attribute that is referenced through the **relating_method** attribute for one or more **action_method_relationship** of type **manufacturing_-process_relationship** that shall reference through the **related_method** attribute either a **manufacturing_-process** or a **non_machining_process**.

**WR9:** The **action_assignment** of type **applied_action_assignment** through **assigned_action** shall references zero or one **feature_dependency** through **items**.

## 5.2.3.2.70 production_rate

A **production_rate** is an **action_property** which is the time it takes to produce a measurable amount of work.

EXPRESS specification:

```
*)
ENTITY production_rate
   SUBTYPE OF (action_property);
WHERE

 (* production_rate.time_per_unit *)

 wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF
   (it)) = 2) AND (it.name = 'time per unit')))) = 1))) = 1;
```

```
 (* production_rate.unit_quantity *)

 wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'unit quantity'))))=1)))=1;

 wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
   (SIZEOF(QUERY ( it <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'production rate source'))))=1)))=1;

  wr4: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
         'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
         (SIZEOF(QUERY ( it <* apr.representation.items |
         (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
         IN TYPEOF(it)) AND (it.name = 'allowed type'))))<=1)))=0));


END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **tepresentation_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'.

**WR4:** Exactly one **representation_item** used for the representation of an **production_rate** shall be of type **descriptive_representation_item** with a name of 'allowed type'.

## 5.2.3.2.71 range_of_parts

A **range_of_parts** is an **action_property** which is the minimum and maximum number of parts that can be manufactured using the process plan.

 

EXPRESS specification:

```
*)
ENTITY range_of_parts
   SUBTYPE OF (action_property);
WHERE
 wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   (SIZEOF(QUERY ( rep <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM'
   IN TYPEOF(rep)) AND (rep.name = 'low value')))) = 1))) = 1;

 wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY') |
   (SIZEOF(QUERY ( rep <* apr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM'
   IN TYPEOF(rep)) AND (rep.name = 'high value')))) = 1))) = 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** Exactly one **representation_item** used for the representation of an **range_of_parts** shall be of type **measure_representation_item** with a name of 'low_value'.

**WR2:** Exactly one **representation_item** used for the representation of an **range_of_parts** shall be of type **measure_representation_item** with a name of 'high_value'.

## 5.2.3.2.72 resource_with_material

the **resource_with_material** is a type of **generic_manufacturing_resource** that defines an additional information for **material**.

EXPRESS specification:

```
*)
ENTITY resource_with_material
   SUBTYPE OF (generic_manufacturing_resource);
WHERE

 wr1:(NOT(SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')|
   ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN TYPEOF(ar)))|
   (NOT (SIZEOF(QUERY ( pdp_imp <* QUERY ( pdp <* arr.OPERATIONS|
   ('PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(pdp)))|
   (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
   ((('PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.description = 'resource material')) AND
   ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION'
   IN TYPEOF(ppa.property_or_shape.definition)))))
   = 1))) = 0)))) = 0));
END_ENTITY;
(*
```

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**.

## 5.2.3.2.73 resource_with_representation

the **resource_with_material** is a type of **generic_manufacturing_resource** that defines an additional information for external shape representation.

EXPRESS specification:

```
*)
ENTITY resource_with_representation
   SUBTYPE OF (generic_manufacturing_resource);
WHERE



 wr1:(NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE')|
   (SIZEOF(QUERY ( rpr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
 ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS'
   IN TYPEOF(rpr.representation)))) = 1)))=0));
END_ENTITY;
(*
```

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**.

## 5.2.3.2.74 revision

The **revison** is a type of **action_relationship** that relates together different versions of a process plan.

EXPRESS specification:

```
*)
ENTITY revision
   SUBTYPE OF (action_relationship);
WHERE
 (* revision.reason_for_revision *)

 wr1: SIZEOF(QUERY ( rc <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.RELATIONSHIP_CONDITION.APPLICABLE_RELATIONSHIPS')|
    (rc.name = 'reason for revision'))) = 1;

 (* revision.revision_level *)
```

 

```
 wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF.related_action.chosen_method,
   'PROCESS_PLANNING_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
   (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST'
   IN TYPEOF(ars.request)) AND
   (ars.request.description = 'revision level')))) = 1;

 (* revision to status_authority *)

 wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS')|
   ('PROCESS_PLANNING_SCHEMA.APPROVAL'
   IN TYPEOF(adr.assigned_approval)))) >= 1;

 (* revision to process_plan_version *)

 wr4: ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION'
   IN TYPEOF(SELF.related_action)) AND
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION'
   IN TYPEOF(SELF.relating_action));
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference to **relationship_condition** through **applicable_relationships** with a name of 'reason for revison'.

**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**.

## 5.2.3.2.75 machining_process

The **machining_process** is a type of **single_setup_activity** that defines the type of machine, machine setup, and activity required to perform a automated machining process.

EXPRESS specification:

```
*)
ENTITY machining_process
   SUBTYPE OF (manufacturing_process);
WHERE

 wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
   'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) )) >= 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 **manufacturing_process_relationship** that references through the **related_-method** an **machine**.

## 5.2.3.2.76 manufacturing_process

A **manufacturing_process** is a type of **action_method** that defines a setup for a specific machine shall perform manufacturing activities necessary to create a part shape.

EXPRESS specification:

```
*)
ENTITY manufacturing_process

   SUBTYPE OF (action_method);
WHERE

 wr1:SIZEOF(QUERY ( amr <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')|
(('PROCESS_PLANNING_SCHEMA.SINGLE_ACTIVITY_RELATIONSHIP' IN TYPEOF (amr)) AND
  (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MACHINE_SETUP',
      'PROCESS_PLANNING_SCHEMA.TOOL_SETUP',
      'PROCESS_PLANNING_SCHEMA.FIXTURE_SETUP',
      'PROCESS_PLANNING_SCHEMA.ANCILLARY_SETUP'] *
     TYPEOF(amr.related_method))=1))))>=1;

 wr2: (SELF.purpose IN ['legacy nc',
           'non nc',
           'integrated nc']);



 wr3: SIZEOF(QUERY ( cm <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
   (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
   (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.name = 'assigned feature')) )) = 1)) )) >= 0;

 wr4: SIZEOF(QUERY ( sar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD') |
   ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY_RELATIONSHIP'
   IN TYPEOF(sar)) AND
   ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY'
   IN TYPEOF(sar.related_method))))>=1;

 wr5: SIZEOF(QUERY ( cm <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
   (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
   ((('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.description = 'intermediate shape')) )))
    >= 0)))) = 0;
```

 

```
   wr6: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
     ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
     IN TYPEOF(arr)) ) |
     (NOT (SIZEOF(QUERY ( rp <* USEDIN(am,
     'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      ('PROCESS_PLANNING_SCHEMA.PART_HOLDING_POSITION' IN TYPEOF(rp))))
     = 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** either a **machine_setup**, **tool_setup**, **fixture_setup**, or **ancillary_setup**.

**WR2:** The **purpose** attribute shall be one of 'legacy nc', 'non nc' or 'integrated nc'.

**WR3:** There shall be zero or more references through the **chosen_method** attribute of an **action** that has a referenced through the **process** attribute of a **process_property_association** that references through the **property_or_shape** attribute to a **shape_aspect** with a **name** of 'assigned feature'.

**WR4:** There shall be one or more references through the **relating_method** attribute of an **action_method_relationship** of type **manufacturing_activity_relationship** that references a **manufacturing_activity** through the **related_method** attribute.

**WR5:** There shall be zero or more references through the **chosen_method** attribute of an **action** that has a referenced through the process attribute of a **process_property_association** that references through the **property_or_shape** attribute to a **shape_aspect** with a description of 'intermediate shape'.

**WR6:** There shall be exactly zero, one, or many reference from a **requirement_for_action_resource** through **resources** that is referenced through **resources** by **resource_property** of kind **part_holding_position**.

## 5.2.3.2.77 manufacturing_process_relationship

The **manufacturing_process_relationship** is a type of **sequential_method** that relates a **manufacturing_process** to either a **machining_process**.

EXPRESS specification:

```
*)
ENTITY manufacturing_process_relationship

   SUBTYPE OF (sequential_method);

WHERE

 wr1: ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
```

```
   IN TYPEOF(SELF.related_method));

 wr2:   ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
   IN TYPEOF(SELF.relating_method));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **related_method** attribute shall be **manufacturing_process**.

**WR2:** The **relating_method** attribute shall be **manufacturing_process**.

## 5.2.3.2.78 single_activity_relationship

The **single_activity_relationship** is a type of **sequential_method** that relates a **manufacturing_process** to a type of setup.

EXPRESS specification:

```
*)
ENTITY single_activity_relationship

   SUBTYPE OF (sequential_method);

WHERE

 wr2: ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS'
   IN TYPEOF(SELF.relating_method));

 wr3:  (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MACHINE_SETUP',
      'PROCESS_PLANNING_SCHEMA.TOOL_SETUP',
      'PROCESS_PLANNING_SCHEMA.FIXTURE_SETUP',
      'PROCESS_PLANNING_SCHEMA.ANCILLARY_SETUP'] *
     TYPEOF(SELF.related_method))=1);
END_ENTITY;
(*
```

Formal propositions:

**WR1:** The **relating_method** attribute shall be **manufacturing_process**.

**WR2:** The **related_method** attribute shall be either a **machine_setup**, **tool_setup**, **fixture_setup**, or **ancillary_setup**

## 5.2.3.2.79 tool_assembly

The **tool_assembly** is a type of **action_resource** that is the tooling to machine the part.

 

EXPRESS specification:

```
*)
ENTITY tool_assembly
   SUBTYPE OF (action_resource);
WHERE

 (* tool_assembly to tool_contract *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT.ITEMS')|
   ('PROCESS_PLANNING_SCHEMA.TOOL_CONTRACT'
   IN TYPEOF(adr.assigned_contract)))) <= 1;

 (* tool_assembly to Tool_placement_instruction *)

 wr2: SIZEOF(QUERY (ar <*(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES') |
    ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT'
    IN TYPEOF(adr)))|
    SIZEOF(QUERY ( rp <* (USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE')) |
('PROCESS_PLANNING_SCHEMA.TOOL_PLACEMENT_INSTRUCTION' IN TYPEOF(rp)))) = 1
    ))
    >=0;

 (* tool_assembly to property_parameter *)

 wr3: SIZEOF(QUERY ( rp <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
   IN TYPEOF(pdr.representation))) |
   (NOT(SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
   TYPEOF(it)) = 1) AND
   (it.name = 'tool assembly tool parameters'))
   )) >= 0)))) = 0)))) = 0;

   wr4: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATING_RESOURCE')
        | ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP' IN
        TYPEOF(arr)) ) | (
        'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT' IN TYPEOF(
        ar2.related_resource)) )) >= 1;

 wr5: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE'
   IN TYPEOF(adr.assigned_document)))))<=1;

   wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES') |
  ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN TYPEOF(ar)) ) |
      (NOT (SIZEOF(QUERY ( pdp_imp <* QUERY ( pdp <* arr.OPERATIONS |
  ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS' IN TYPEOF(pdp)) ) |
      (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
```

```
      'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
(('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.property_or_shape)) AND
      (ppa.description = 'tool assembly shape'))
      )) >= 0) )) = 0)) )) = 0;
  wr7: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
    (NOT(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
 ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.representation))) |
    ((impl_rep.representation.name = 'tool assembly parameter')AND
    (SIZEOF(QUERY ( it <* impl_rep.representation.items |
  ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF(it)) AND
    ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE'
    IN TYPEOF(it\measure_with_unit.value_component)) AND
    (it.name = 'number of spares')
    ))<=1)) ))=0)))) =0));

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall one or more references from a **contract_assignment** of kind **applied_contract_-assignment** through the **items** attribute that references a **tool_contract** through the **assigned_contract**.

**WR2:** There shall be exactly zero, one, or many reference from a **requirement_for_action_resource** through **resources** that is referenced through **resources** by **resource_property** of kind **tool_-placement_instruction**.

**WR3:** There shall be one or more reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **measure_representation_-item** and **descriptive_representation_item** with a **name** of 'tool assembly usage specification'

**WR4:** There shall be zero or one **action_resource** referenced by **resource** that is referenced by an **action_resource_relationship** through **related_resource** that references a **tool_assembly_element** through **relating_resource**

**WR5:** There shall one or more references from an **applied_document_reference** through the **items** attribute that references a **design_reference** through the **assigned_contract**.

**WR6:** There shall be a reference through the **property** attribute a **resource_property** that shall reference through the **resource** attribute a **action_resource_requirement** that shall reference through the **operations** attribute a **property_process** that is referenced through the **process** attribute by a **process_-property_association** with a description of 'tool assembly shape' that references through the **property_-or_shape** attribute a **shape_aspect**.

**WR7:** There shall be at most one reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **representation** with a **name** of 'machine usage specification' that references through the **items** attribute the **measure_-representation_item** and **count_measure** with a **name** of 'number of spares'.

 

## 5.2.3.2.80 tool_assembly_element

The **tool_assembly_element** is a type of **action_resource** that is used to define portions of the **tool_-assembly**.

EXPRESS specification:

```
*)
ENTITY tool_assembly_element
   SUBTYPE OF (action_resource);
WHERE

 (* tool_assembly_element.catalogue_number *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (adr.assigned_document.description ='tool catalogue number')
   )) <= 1;

 (* tool_assembly_element.company_name *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')|
   (adr.assigned_organization.description = 'tool company name')
   )) = 1;

 (* tool_assembly_element.id *)

 wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (adr.description = 'tool element identification'))) = 1;


 (* tool_assembly_element to specification *)


   wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
        | ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
        TYPEOF(adr.assigned_document)) )) >= 0;


 (* tool_assembly_element to
externally_defined_representation_with_parameters *)

 wr5: (NOT(SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY') |
   ('PROCESS_PLANNING_SCHEMA.'+
   'EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS'
   IN TYPEOF(apr)))) <= 1))) = 0));
```

```
 wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')|
   ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT'
   IN TYPEOF(ar))) | (NOT (SIZEOF(QUERY ( pdp_imp <*
   QUERY ( pdp <* arr.operations |
   ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS'
   IN TYPEOF(pdp))) | (SIZEOF(QUERY ( ppa <*
   USEDIN(pdp_imp,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')|
   ((('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.description = 'tool assembly element shape'))))) =
   0))) = 0)))) >= 0;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall exactly zero or one references from a **applied_document_assignment** through the **items** attribute that references a **document** with a **description** of "tool catalogue number' of through the **assigned_document**.

**WR2:** There shall exactly one references from a **applied_organization_assignment** through the **items** attribute that references a **organization** with a **description** of 'tool company name' of through the **assigned_organization**.

**WR3:** There shall be exactly one reference from a **resource_property** through **resource** with a **description** of 'tool element identification'.

**WR4:** The **tool_assembly_element** shall be in the set of **items** of exactly one **applied_document_reference** with a **assigned_document** defines the **process_plan_specification** containing the specification.

**WR5:** There shall zero or 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**.

**WR6:** There shall be a reference through the **property** attribute a **resource_property** that shall reference through the **resource** attribute a **action_resource_requirement** that shall reference through the **operations** attribute a **property_process** that is referenced through the **process** attribute by a **process_-property_association** with a description of 'tool assembly shape' that references through the **property_-or_shape** attribute a **shape_aspect**.

## 5.2.3.2.81 tool_body

A **tool_body** is a type of **tool_assembly** that represents the main portion of the tool assembly.

 

EXPRESS specification:

```
*)
ENTITY tool_body
   SUBTYPE OF (tool_assembly_element);
WHERE
 wr1: SIZEOF(QUERY ( rp <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   (NOT(SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
   IN TYPEOF(pdr.representation))) |
   (NOT(SIZEOF(QUERY ( it <* impl_rep.representation.items |
   ((SIZEOF([
   'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
   'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
   TYPEOF(it)) = 1) AND
   (it.name = 'tool body parameter'))
   )) >= 0)))) = 0)))) = 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be one or more reference from a **resource_property** through **resource** that is referenced by **resource_property_representation** through the **property** that references a **measure_representation_item** and **descriptive_representation_item** with a **name** of 'tool body parameters'

## 5.2.3.2.82 tool_contract

A **tool_contract** is a type of **contract** that defines documentation for machine tools.

EXPRESS specification:

```
*)
ENTITY tool_contract
   SUBTYPE OF (contract);
WHERE

   (* tool_contract.design_order *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST'
   IN TYPEOF(adr.assigned_action_request)) AND
   (adr.assigned_action_request.description = 'design order')
   ))) =  1;

   (* tool_contract.fabrication_order *)

 wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')|
   (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST'
   IN TYPEOF(adr.assigned_action_request)) AND
   (adr.assigned_action_request.description = 'fabrication order')
   ))) = 1;
```

```
   (* tool_contract to document_assignment*)

 wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')|
   (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
   IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items |
   (('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN TYPEOF(it))
   AND (it\document.description = 'tool contract document')))) <= 1
   ))))) = 0;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall exactly one references from a **applied_action_request_assignment** of kind **actio_-request_assignment** through the **items** attribute that references a **versioned_action_request** with a **name** of 'design order' through the **assigned_action_request**.

**WR2:** There shall exactly one references from a **applied_action_request_assignment** of kind **actio_-request_assignment** through the **items** attribute that references a **versioned_action_request** with a **name** of 'fabrication order' through the **assigned_action_request**.

**WR3:** There shall exactly one references from a **applied_action_request_assignment** of kind **actio_-request_assignment** through the **items** attribute that references through the **assigned_action_request** a **versioned_action_request**, that is referenced by exactly one **action_directive** with a name of 'fixture contract' through **requests** that is referenced by an **action_resource** with a **name** of 'part number' through **usage**.

## 5.2.3.2.83 tool_placement_instruction

The **tool_placement_instruction** is a type of **resource_property** that is additional information for a machine tool.

EXPRESS specification:

```
*)
ENTITY tool_placement_instruction
   SUBTYPE OF (resource_property);
WHERE

 (* Tool_placement_instruction.tool_position*)

 wr1: SIZEOF(QUERY ( rpr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
   ((rpr.name = 'tool position') AND
   (SIZEOF(QUERY ( it <* rpr.representation.items |
   (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
   IN TYPEOF(it)) AND (it.name = 'tool position')
   ))) = 1)))) =  0;

 (* Tool_placement_instruction to tool_magazine_turret_carousel*)

 wr2: ('PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL' IN
    TYPEOF(SELF.resource));
```

 

```
 (* Tool_placement_instruction to specification *)


 wr3: SIZEOF(QUERY (adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN TYPEOF
   (adr.assigned_document))))=1;
END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference by an **action_property** through **definition** that is referenced by an **action_property_representation** through **property** that references **representation** through **representation** that references a **descriptive_representation_item** with a **name** of 'tool position'.

**WR2:** There shall be zero or one **action_resource** referenced by **resource** that is referenced by an **action_resource** that is a **tool_magazine_turret_carousel** through **relating_resource**

**WR3:** The **tool_placement_instruction** 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.

## 5.2.3.2.84 tool_setup

A **tool_setup** is an **action_method** which defines the machine, and machine elements where the tool is intended to be used.

EXPRESS specification:

```
*)
ENTITY tool_setup
   SUBTYPE OF (process_plan_activity);
WHERE

  (* tool_setup to machine *)

   wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
     ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
     IN TYPEOF(arr)) ) |
     (NOT (SIZEOF(QUERY ( it <* am.resources |
     ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) ))
     = 1) )) ) = 0;

  (* tool_setup to tool_assembly *)

   wr2: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (NOT (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)) )) =
      1)) )) = 0;
```

```
  (* tool_setup to tool_magazine_turret_carousel *)

   wr3: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
      ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
      IN TYPEOF(arr)) ) |
      (NOT (SIZEOF(QUERY ( it <* am.resources |
      ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)) )) =
      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 a **machine** through **resources**.

**WR2:** 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 **tool_assembly**s through **resources**.

**WR3:** 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 **tool_magazine_turret_carousel** through **resources**.

## 5.2.3.2.85 tool_assembly_relationship

A **tool_assembly_relationship** is an **action_resource_relationships** which relates a tool assembly to a tool assembly element.

EXPRESS specification:

```
*)
ENTITY tool_assembly_relationship
   SUBTYPE OF (action_resource_relationship);
WHERE
   wr1: (('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(SELF.
        relating_resource)) AND (
        'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT' IN TYPEOF(
        SELF.related_resource))) OR ((
        'PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL' IN
        TYPEOF(SELF.relating_resource)) AND (
        'PROCESS_PLANNING_SCHEMA.WORKSTATION' IN TYPEOF(SELF.
        related_resource)));

END_ENTITY;
(*
```

 

Formal proposition:

**WR1**: The **relating_resource** is a **fixture_assembly**.

**WR2**: The **related_resource** is a **tool_assembly**, **tool_assembly_element, magazine_turret_carousel** or **workstation**.

## 5.2.3.2.86 tool_magazine_turret_carousel

A **tool_magazine_turret_carousel** is an **action_resource** which defines the holder of the tool assemblies for machining operations

EXPRESS specification:

```
*)
ENTITY tool_magazine_turret_carousel
   SUBTYPE OF (action_resource);
WHERE

 (* tool_magazine_turret_carousel.tool_capacity *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (adr
   .description = 'tool capacity'))) = 1;

 (* tool_magazine_turret_carousel to workstation *)

 wr2: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.' +
   'RELATING_RESOURCE') | (
   'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP' IN
   TYPEOF(arr))) | ('PROCESS_PLANNING_SCHEMA.WORKSTATION' IN
   TYPEOF(ar2.related_resource)))) = 1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference from a **resource_property** through **resource** with a **description** of 'tool capacity'.

**WR2:** There shall be one or more references from a **action_resource_relationship** of type **tool_-assembly_relationship** through **relating_resource** that references a **workstation** through **related_-resource**.

## 5.2.3.2.87 validation

A **validation** is an **action_method** which is the potential technique of verifying product requirements.

EXPRESS specification:

```
*)
ENTITY validation
   SUBTYPE OF (manufacturing_activity);
WHERE

(* validation to fixture_assembly *)

 wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
   ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
   IN TYPEOF(arr))) |
   (SIZEOF(QUERY ( it <* am.resources |
   ('PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(it))))
   <= 1))) = 1;

 (* validation to tool_assembly *)

 wr2: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
   ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
   IN TYPEOF(arr))) |
   (SIZEOF(QUERY ( it <* am.resources |
   ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)))) =
   1))) = 1;

 (* validation to part_shape *)

 wr3: SIZEOF(QUERY ( cm <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') |
   (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
   'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS') |
   (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE'
   IN TYPEOF(ppa.property_or_shape)) AND
   (ppa.description = 'part fixture mounting part shape')))) = 1)))) = 1;

 (* validation to controller_program *)

 wr4: SIZEOF(QUERY ( adr <*
   SELF\action_method_with_associated_documents.documents |
   ('PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM' IN TYPEOF(adr)
   ))) <=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 **fixture_assembly**s through **resources**.

**WR2:** 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 **tool_assembly**s through **resources**.

**WR3:** There shall be exactly one reference by an **action** through **chosen_method** that is referenced by

 

a **process_property_association** through **process** with a **description** of 'part fixture mounting part shape' that references **product_definition_shape** through **property_or_shape**.

**WR4:** There shall be exactly one reference by an **action** through **chosen_method** that is referenced by a **process_property_association** through **process** with a **description** of 'part fixture mounting part shape' that references **product_definition_shape** through **property_or_shape**.

## 5.2.3.2.88 value_range

A **value_range** is a type of **compound_representation_item** that specifies a range of values defined by two measure_representation_items.

EXPRESS specification:

```
*)
ENTITY value_range
  SUBTYPE OF (compound_representation_item);

WHERE

wr1: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
   ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
    ))=2;

wr2: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
 (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri)) AND
   (mri.name='lower limit'))
    ))=1;

wr3: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
 (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri)) AND
   (mri.name='upper limit'))
    ))=1;

wr4: SIZEOF(QUERY( i1 <* SELF.item_element |
 ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i1)) AND
   (SIZEOF ( QUERY (i2 <* SELF.item_element |
 ('PROCESS_PLANNING_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**.

## 5.2.3.2.89 view_reference

A view_reference is a type of document_usage_constraint that is identification, within a drawing, of a specific area of interest.

EXPRESS specification:

```
*)
ENTITY view_reference
   SUBTYPE OF (document_usage_constraint);
WHERE

  (* view_reference to activity *)

   wr1: (NOT(SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF.source,
      'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')|
  ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN TYPEOF(dr)) ) |
      (SIZEOF(QUERY ( d <* adr.items |
      ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) ))
      >= 0) )) = 0));

 wr2: 'PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF (SELF.source);

 wr3: SELF.subject_element IN ['sheet','view','zone'];

END_ENTITY;
(*
```

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'.

## 5.2.3.2.90 work_cell

 

A **work_cell** is a **group_assignment** which is a collection of **action_resource**s.

EXPRESS specification:

```
*)
ENTITY work_cell
   SUBTYPE OF (applied_group_assignment,action_resource);
WHERE

 (* work_cell to workstation *)

 wr1: SIZEOF(QUERY ( adr <* SELF.items |
   ('PROCESS_PLANNING_SCHEMA.WORKSTATION'
   IN TYPEOF(adr)))) >= 1;

 (* work_cell to activity *)

 wr2: SIZEOF(QUERY ( adr <* SELF.usage |
   ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY'
   IN TYPEOF(adr)))) >= 0;

 wr3: SIZEOF(QUERY( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
   ('PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION') IN TYPEOF (ar)))=1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one **workstation** in the list of **items**.

**WR2:** There shall be zero, one or many **work_cell** references through **usage** a **process_plan_activity**.

**WR3:** There shall be exactly one reference from a **resource_property** through **resource** of type **in_-facility_location**.

## 5.2.3.2.91 workstation

A **workstation** is a type of **action_resource** that is identification of an area where work .

EXPRESS specification:

```
*)
ENTITY workstation
   SUBTYPE OF (action_resource);
WHERE

 (* workstation to special_capability *)

 wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
   (adr.assigned_document.kind.product_data_type='special capability')))>=0;
```

```
 WR2: SIZEOF(QUERY(wc <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT.ITEMS') |
   ('PROCESS_PLANNING_SCHEMA.WORK_CELL') IN TYPEOF (wc)))=1;

END_ENTITY;
(*
```

Formal propositions:

**WR1:** There shall be exactly one reference from a **applied_document_reference** through **items** of type **document_reference** that references through **assigned_document** a **document** that references through **kind** a **document_type** with the.attribute **product_data_type** of value 'special capability'.

**WR2:** There shall be exactly one reference from a **applied_group_assignemt** through **items** of type **group_assignment** that is of type wor**k_cell.**

## 5.2.3.3 process planning schema imported entity modifications

## 5.2.3.3.1 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 5.2.3.4.1).

## 5.2.3.3.2 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 5.2.3.4.1).

## 5.2.3.3.3 action_resource

The base definition of the **action_resource** 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_resource** entity:

— subtype_exclusiveness_action_resource (see 5.2.3.7.1).

### 5.2.3.3.4 action_resource_relationship

The base definition of the **action_resource_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_resource_relationship** entity:

— subtype_exclusiveness_action_resource_relationship (see 5.2.3.7.2).

### 5.2.3.3.5 approval

The base definition of the **approval** 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 **approval** entity:

— approval_requires_approval_date_time (see 5.2.3.4.2).

— approval_requires_approval_person_organization (see 5.2.3.4.3).

### 5.2.3.3.6 approval_date_time

The base definition of the **approval_date_time** 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 **approval_date_time** entity:

— approval_requires_approval_date_time (see 5.2.3.4.2).

### 5.2.3.3.7 approval_person_organization

The base definition of the **approval_person_organization** 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 **approval_person_organization** entity:

— approval_requires_approval_person_organization (see 5.2.3.4.3).

### 5.2.3.3.8 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 rule defined in this part of ISO 10303 applies to the **approval_status** entity:

— dependent_instantiable_approval_status (see 5.2.3.4.4).

### 5.2.3.3.9 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_exclusiveness_characterized_object (see 5.2.3.7.3).

### 5.2.3.3.10 context_dependent_action_method_relationship

The base definition of the **context_dependent_action_method_relationship** entity is given in ISO 10303-49. 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 **context_dependent_action_-method_relationship** entity:

— subtype_exclusiveness_context_dependent_action_method_relation (see ?).

### 5.2.3.3.11 contract

The base definition of the **contract** 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 **contract** entity:

— subtype_exclusiveness_contract (see 5.2.3.7.4).

### 5.2.3.3.12 date

The base definition of the **date** 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 **date** entity:

— dependent_instantiable_date (see 5.2.3.4.5).

### 5.2.3.3.13 document

The base definition of the **document** 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 **document** entity:

— subtype_exclusiveness_document (see 5.2.3.7.5).

### 5.2.3.3.14 externally_defined_class

The base definition of the **externally_defined_class** 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 **externally_defined_class** entity:

— externally_defined_class_with_known_source_requirement (see 5.2.3.4.12).

### 5.2.3.3.15 geometric_tolerance

The base definition of the **geometric_tolerance** entity is given in ISO 10303-47. 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 **geometric_tolerance** entity:

geometric_tolerance_subtype_exclusiveness (see 5.2.3.4.13).

### 5.2.3.3.16 group

The base definition of the **group** 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 **group** entity:

— subtype_exclusiveness_group (see 5.2.3.7.6).

### 5.2.3.3.17 instanced_feature

The base definition of the **instanced_feature** entity is given in ISO 10303-521. 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 **instanced_feature** entity:

— machining_feature_life_cycle (see 5.2.3.4.14).

### 5.2.3.3.18 named_unit

The base definition of the **named_unit** 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 **named_unit** entity:

— dependent_instantiable_named_unit (see 5.2.3.4.6).

### 5.2.3.3.19 precision_qualifier

The base definition of the **precision_qualifier** 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 **precision_qualifier** entity:

— dependent_instantiable_precision_qualifier (see 5.2.3.4.7).

### 5.2.3.3.20 product_definition_relationship

The base definition of the **product_definition_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 **product_definition_relationship** entity:

— mating_definition_relationship_orientation (see 5.2.3.5.1).

### 5.2.3.3.21 resource_property

The base definition of the **resource_property** entity is given in ISO 10303-49. 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 **resource_property** entity:

— subtype_exclusiveness_resource_property (see 5.2.3.7.7).

### 5.2.3.3.22 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 rule defined in this part of ISO 10303 applies to the **security_classification_level** entity:

— dependent_instantiable_security_classification_level (see 5.2.3.4.10).

### 5.2.3.3.23 shape_aspect

The base definition of the **shape_aspect** 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 **shape_aspect** entity:

— subtype_exclusiveness_shape_aspect (see 5.2.3.7.8).

### 5.2.3.3.24 shape_representation

The base definition of the **shape_representation** 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 **shape_representation** entity:

— subtype_exclusiveness_shape_representation (see 5.2.3.7.9).

### 5.2.3.3.25 type_qualifier

The base definition of the **type_qualifier** 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 **type_qualifier** entity:

— dependent_instantiable_type_qualifier (see 5.2.3.4.8).

### 5.2.3.3.26 uncertainty_qualifier

The base definition of the **uncertainty_qualifier** 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 **uncertainty_qualifier** entity:

— dependent_instantiable_uncertainty_qualifier (see 5.2.3.4.9).

## 5.2.3.4 Process planning schema rules

### 5.2.3.4.1 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 =
   'process_planning_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 'process_planning_schema' as its **application_interpreted_model_schema_name**.

## 5.2.3.4.2 approval_requires_approval_date_time

The **approval_requires_approval_date_time** rule specifies that each instance of **approval** shall be referenced by exactly one **approval_date_time**. This rule enforces the requirement for every approval to have a date on which the approval obtained its specified status.

EXPRESS specification:

```
*)
RULE approval_requires_approval_date_time FOR (approval,
 approval_date_time);
WHERE
  wr1: SIZEOF(QUERY ( app <* approval |
     (NOT (SIZEOF(QUERY ( adt <*
     approval_date_time | (app :=: adt.dated_approval))) = 1))))
     = 0;
END_RULE;
(*
```

Argument definitions:

**approval:** the set of all instances of **approval** entities.

**approval_date_time:** the set of all instances of **approval_date_time** entities.

Formal propositions:

**WR1:** For each instance of **approval**, there shall be exactly one instance of **approval_date_time** which contains the instance of **approval** as its **dated_approval** attribute.

## 5.2.3.4.3 approval_requires_approval_person_organization

The **approval_requires_approval_person_organization** specifies that each instance of **approval** shall have at least one **approval_person_organization** referencing it. This rule enforces the requirement for an approval to be authorized by one or more people within their organizations.

EXPRESS specification:

```
*)
RULE approval_requires_approval_person_organization FOR
 (approval, approval_person_organization);
WHERE
  wr1: SIZEOF(QUERY ( app <* approval |
     (NOT (SIZEOF(QUERY ( apo <*
     approval_person_organization |
     (app :=: apo.authorized_approval))) >= 1)))) = 0;
END_RULE;
(*
```

Argument definitions:

**approval:** the set of all instances of **approval** entities.

**approval_person_organization:** the set of all instances of **approval_person_organization** entities.

Formal propositions:

**WR1:** For each instance of **approval**, there shall be one or more instances of **approval_person_-organization** which contains the instance of **approval** as its **authorized_approval** attribute.

## 5.2.3.4.4 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.

## 5.2.3.4.5 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.

## 5.2.3.4.6 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**.

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.

## 5.2.3.4.7 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.

## 5.2.3.4.8 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.

## 5.2.3.4.9 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.

## 5.2.3.4.10 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.

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.

## 5.2.3.4.11 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.

## 5.2.3.4.12 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 |
     (('PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE'
  IN TYPEOF(edc.source)) AND (NOT (SIZEOF(QUERY ( aoa <* USEDIN(edc.source,
     'PROCESS_PLANNING_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'.

 

## 5.2.3.4.13 geometric_tolerance_subtype_exclusiveness

The **geometric_tolerance_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **geometric_tolerance** shall be only one of **angularity_tolerance**, **circular_runout_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **line_profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **roundness_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance**.

EXPRESS specification:

```
*)
RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);
WHERE
 WR1: SIZEOF (QUERY (gt <* geometric_tolerance |
    NOT (SIZEOF (TYPEOF (gt) *
    ['PROCESS_PLANNING_SCHEMA.ANGULARITY_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.CIRCULAR_RUNOUT_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.CONCENTRICITY_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.CYLINDRICITY_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.FLATNESS_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.LINE_PROFILE_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.PARALLELISM_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.PERPENDICULARITY_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.POSITION_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.ROUNDNESS_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.STRAIGHTNESS_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.SURFACE_PROFILE_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.SYMMETRY_TOLERANCE',
    'PROCESS_PLANNING_SCHEMA.TOTAL_RUNOUT_TOLERANCE'])
    <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

**geometric_tolerance:** the set of all instances of **geometric_tolerance** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **geometric_tolerance** shall be one of **angularity_tolerance**, **circular_runout_tolerance**, **concentricity_tolerance**, **cylindricity_tolerance**, **flatness_tolerance**, **line_-profile_tolerance**, **parallelism_tolerance**, **perpendicularity_tolerance**, **position_tolerance**, **roundness_tolerance**, **straightness_tolerance**, **surface_profile_tolerance**, **symmetry_tolerance**, or **total_runout_tolerance**.

## 5.2.3.4.14 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);
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'.

### 5.2.3.5 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, 'PROCESS_PLANNING_SCHEMA.' +
    'MATERIAL_DESIGNATION.DEFINITIONS')) = 0) |
    NOT (SIZEOF (QUERY (mfuo <* make_from_usage_option |
    NOT (nmpd :=: mfuo.relating_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 **relating_product_definition**.

 

### 5.2.3.5.1 mating_definition_relationship_orientation

The **mating_definition_relationship_orientation** rule specifies that each instance of a **product_-definition_relationship** may or may not have an orientation.

EXPRESS specification:

```
*)
RULE mating_definition_relationship_orientation FOR
  (product_definition_relationship);

WHERE

  wr1: SIZEOF(QUERY ( prd_def_rel <* product_definition_relationship |
       (NOT (SIZEOF(QUERY ( pd <* USEDIN(prd_def_rel,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
   'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')|
       ((('PROCESS_PLANNING_SCHEMA.SHAPE_DEFINITION_REPRESENTATION'
       IN TYPEOF(pdr)) AND
       ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION'
       IN TYPEOF(pdr.used_representation))) AND
       (NOT(SIZEOF(QUERY ( rm <* USEDIN(pdr.used_representation,
     'PROCESS_PLANNING_SCHEMA.REPRESENTATION_MAP.MAPPED_REPRESENTATION')|
       (SIZEOF(QUERY ( mi <* USEDIN(rm,
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM.MAPPING_SOURCE') |
       ('PROCESS_PLANNING_SCHEMA.REPRESENTATION_ITEM'
       IN TYPEOF(mi.mapping_target))))
       <= 1))) = 0)))))=0))))=0))))=0;

END_RULE;
(*
```

Argument definitions:

**product_definition_relationship**: the set of all instances of **product_definition_relationship**.

Formal propositions:

**WR1**: Each instance **product_definition_relatonship** that defines the part placement for a **mating_definition_relationship** has zero or one orientation defined by a **mapped_item**

### 5.2.3.5.2 restrict_name_for_known_source

The **restrict_name_for_known_source** rule ensures that each instance of **known_source** has a name of 'ISO 13584 library'.

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
(*
```

Argument definitions:

**known_source**: the set of all instances of **known_source**.

Formal propositions:

**WR1**: Each instance of **known_source** shall have a name of 'ISO 13584 library' or 'ISO 10303 part'.

## 5.2.3.6 shape_aspect_relationship_subtype_exclusiveness

The **shape_aspect_relationship_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **shape_aspect_relationship** shall be only one of **dimensional_location**, **geometric_tolerance_relationship**, **feature_component_relationship**, or **shape_defining_relationship**.

EXPRESS specification:

```
*)
RULE shape_aspect_relationship_subtype_exclusiveness FOR
 (shape_aspect_relationship);
WHERE
 WR1: SIZEOF (QUERY (sr <* shape_aspect_relationship |
    NOT (SIZEOF (TYPEOF (sr) *
    ['PROCESS_PLANNING_SCHEMA.DIMENSIONAL_LOCATION',
    'PROCESS_PLANNING_SCHEMA.GEOMETRIC_TOLERANCE_RELATIONSHIP',
    'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP',
    'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP'])
    <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

**shape_aspect_relationship:** the set of all instances of **shape_aspect_relationship** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **shape_aspect_relationship** shall be one of the **dimensional_location**, **geometric_tolerance_relationship**, **feature_component_relationship**, or **shape_defining_relationship**.

 

### 5.2.3.7 shape_representation_subtype_exclusiveness

The **shape_representation_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **shape_representation** shall be only one of **advanced_brep_shape_representation**, **shape_-representation_with_parameters**, or **shape_dimension_representation**.

EXPRESS specification:

```
*)
RULE shape_representation_subtype_exclusiveness FOR (shape_representation);
WHERE
 WR1: SIZEOF (QUERY (sr <* shape_representation |
    NOT (SIZEOF (TYPEOF (sr) *
   ['PROCESS_PLANNING_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
   'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
   'PROCESS_PLANNING_SCHEMA.SHAPE_DIMENSION_REPRESENTATION'])
    <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

**shape_representation:** the set of all instances of **shape_representation** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **shape_representation** shall be one of the **advanced_brep_-shape_representation**, **shape_representation_with_parameters**, or **shape_dimension_representation**.

### 5.2.3.7.1 subtype_exclusiveness_action_resource

The **subtype_exclusiveness_action_resource** rule specifies that an instance of the subtypes of a **action_resource** shall be only one of **controller**, **generic_manufacturing_resource**, **fixture_assembly**, **fixture_assembly_element**, **work_cell**, **machine**, **workstation**, **tool_magazine_turret_carousel**, **tool_-assembly** or **tool_assembly_element**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_action_resource
 FOR (action_resource);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action_resource |
    NOT (type_check_function(a,[
    'PROCESS_PLANNING_SCHEMA.CONTROLLER',
    'PROCESS_PLANNING_SCHEMA.GENERIC_MANUFACTURING_RESOURCE',
    'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
    'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT',
    'PROCESS_PLANNING_SCHEMA.WORK_CELL',
    'PROCESS_PLANNING_SCHEMA.MACHINE',
    'PROCESS_PLANNING_SCHEMA.PALLET',
    'PROCESS_PLANNING_SCHEMA.WORKSTATION',
    'PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL',
    'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY',
```

```
    'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT'],3)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**action_resource:** the set of all instances of **action_resource** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **action_resource** shall be one of **controller**, **generic_manufacturing_resource**, **fixture_assembly**, **fixture_assembly_element**, **work_cell**, **machine**, **pallet**, **workstation**, **tool_magazine_turret_carousel**, **tool_assembly** or **tool_assembly_element**.

### 5.2.3.7.2 subtype_exclusiveness_action_resource_relationship

The **subtype_exclusiveness_action_resource_relationship** rule specifies that an instance of the subtypes of a **action_resource_relationship** shall be only one of **fixture_assembly_relationship**, or **tool_assembly_relationship**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_action_resource_relationship
 FOR (action_resource_relationship);
 WHERE
  wr1 : SIZEOF(QUERY(a <* action_resource_relationship |
    NOT (type_check_function(a,[
    'PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP',
    'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_RELATIONSHIP',
    'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP'],3)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**action_resource_relationship :** the set of all instances of **action_resource_relationship** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **action_resource_relationship** shall be one of **fixture_assembly_relationship**, or **tool_assembly_relationship**.

### 5.2.3.7.3 subtype_exclusiveness_characterized_object

The **subtype_exclusiveness_characterized_object** rule specifies that an instance of the subtypes of a **characterized_object** shall be only one of **document_file**, **feature_definition**, or **feature_component_definition**.

 

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_characterized_object FOR (characterized_object);
 WHERE
  wr1 : SIZEOF(QUERY(a <* characterized_object | NOT (type_check_function(a,[
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE',
     'PROCESS_PLANNING_SCHEMA.FEATURE_DEFINITION',
     'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION'],3)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**characterized_object :** the set of all instances of **characterized_object** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **characterized_object** shall be one **document_file**, **feature_-definition**, or **feature_component_definition**.

## 5.2.3.7.4 subtype_exclusiveness_contract

The **subtype_exclusiveness_contract** rule specifies that an instance of the subtypes of a **contract** shall be only one of **fixture_contract**, **part_contract**, or **tool_contract**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_contract
 FOR (contract);
 WHERE
  wr1 : SIZEOF(QUERY(a <* contract |
    NOT (type_check_function(a,[
     'PROCESS_PLANNING_SCHEMA.FIXTURE_CONTRACT',
     'PROCESS_PLANNING_SCHEMA.PART_CONTRACT',
     'PROCESS_PLANNING_SCHEMA.TOOL_CONTRACT'],3)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**contract :** the set of all instances of **contract** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **contract** shall be **fixture_contract**, **part_contract**, or **tool_contract**.

## 5.2.3.7.5 subtype_exclusiveness_document

The **subtype_exclusiveness_document** rule specifies that an instance of the subtypes of a **document** shall be only one of **document_file**, **design_reference**, **controller_program**, or **illustration**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_document FOR (document);
 WHERE
  wr1 : SIZEOF(QUERY(a <* document | NOT (type_check_function(a,[
    'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION',
    'PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE',
    'PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE',
    'PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM',
    'PROCESS_PLANNING_SCHEMA.ILLUSTRATION'],4)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**document:** the set of all instances of **document** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **document** shall be **document_file**, **design_reference**, **controller_program**, or **illustration**.

## 5.2.3.7.6 subtype_exclusiveness_group

The **subtype_exclusiveness_group** rule specifies that an instance of the subtypes of a **group** shall be only one of **class**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_group FOR (group);

 WHERE
  wr1: SIZEOF(QUERY ( a <* group | (NOT type_check_function(a,[
       'PROCESS_PLANNING_SCHEMA.CLASS'],3)) )) = 0;

 END_RULE; -- subtype_exclusiveness_group
(*
```

Argument definitions:

**group:** the set of all instances of **group** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **group** shall be **class**.

## 5.2.3.7.7 subtype_exclusiveness_resource_property

The **subtype_exclusiveness_resource_property** rule specifies that an instance of the subtypes of a **resource_property** shall be only one of **tool_placement_instruction**, or **in_facility_location**.

 

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_resource_property FOR (resource_property);
 WHERE
  wr1 : SIZEOF(QUERY(a <* resource_property | NOT (type_check_function(a,[
     'PROCESS_PLANNING_SCHEMA.PART_HOLDING_POSITION',
     'PROCESS_PLANNING_SCHEMA.TOOL_PLACEMENT_INSTRUCTION',
     'PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION'],3)))) = 0;
 END_RULE;
(*
```

Argument definitions:

**resource_property:** the set of all instances of **resource_property** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **resource_property** shall be **tool_placement_instruction**, or **in_-facility_location**.

## 5.2.3.7.8 subtype_exclusiveness_shape_aspect

The **subtype_exclusiveness_shape_aspect** rule specifies that an instance of the subtypes of a **shape_aspect** shall be one of **path_feature_component**, **slot_end**, **pocket_bottom**, **boss_top**, **hole_-bottom**, **applied_area**, **taper**, **chamfer_offset**, **circular_closed_profile**, **ngon_closed_profile**, **closed_-path_profile**, **square_u_profile**, **tee_profile**, **vee_profile**, **rib_top_floor**, **profile_floor**, **rectangular_-closed_profile**, **partial_circular_profile**, **rounded_u_profile**, **open_path_profile**, **modified_pattern**, **tolerance_zone**, **transition_feature**, **derived_shape_aspect**, **composite_shape_aspect**, **symmetric_-shape_aspect**, **datum**, **datum_feature**, or **datum_target**.

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_shape_aspect FOR (shape_aspect);
 WHERE
  wr1 : SIZEOF(QUERY(a <* shape_aspect | NOT (type_check_function(a,[
     'PROCESS_PLANNING_SCHEMA.APPLIED_AREA',
     'PROCESS_PLANNING_SCHEMA.BOSS_TOP',
     'PROCESS_PLANNING_SCHEMA.CHAMFER_OFFSET',
     'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
     'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE',
     'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT',
     'PROCESS_PLANNING_SCHEMA.DATUM',
     'PROCESS_PLANNING_SCHEMA.DATUM_FEATURE',
     'PROCESS_PLANNING_SCHEMA.DATUM_TARGET',
     'PROCESS_PLANNING_SCHEMA.DERIVED_SHAPE_ASPECT',
     'PROCESS_PLANNING_SCHEMA.HOLE_BOTTOM',
     'PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE',
     'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN',
     'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
     'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE',
     'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
     'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT',
     'PROCESS_PLANNING_SCHEMA.POCKET_BOTTOM',
```

```
    'PROCESS_PLANNING_SCHEMA.PROFILE_FLOOR',
    'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
    'PROCESS_PLANNING_SCHEMA.RIB_TOP_FLOOR',
    'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
    'PROCESS_PLANNING_SCHEMA.SLOT_END',
    'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
    'PROCESS_PLANNING_SCHEMA.SYMMETRIC_SHAPE_ASPECT',
    'PROCESS_PLANNING_SCHEMA.TAPER',
    'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
    'PROCESS_PLANNING_SCHEMA.TOLERANCE_ZONE',
    'PROCESS_PLANNING_SCHEMA.TRANSITION_FEATURE',
    'PROCESS_PLANNING_SCHEMA.VEE_PROFILE'],3)))) = 0;
END_RULE;
(*
```

Argument definitions:

**shape_aspect:** the set of all instances of **shape_aspect** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **shape_aspect** shall be **path_feature_component**, **slot_end**, **pocket_bottom**, **boss_top**, **hole_bottom**, **applied_area**, **taper**, **chamfer_offset**, **circular_closed_profile**, **ngon_closed_profile**, **closed_path_profile**, **square_u_profile**, **tee_profile**, **vee_profile**, **rib_top_floor**, **profile_floor**, **rectangular_closed_profile**, **partial_circular_profile**, **rounded_u_profile**, **open_path_profile**, **modified_pattern**, **tolerance_zone**, **transition_feature**, **derived_shape_aspect**, **composite_shape_aspect**, **symmetric_shape_aspect**, **datum**, **datum_feature**, or **datum_target**.

## 5.2.3.7.9 subtype_exclusiveness_shape_representation

The **subtype_exclusiveness_shape_representation** rule specifies that an instance of the subtypes of a **shape_representation** shall be only one of **advanced_brep_shape_representation**, **direction_shape_representation**, **edge_based_wireframe_shape_representation**, **face_shape_representation**, **faceted_brep_shape_representation**, **geometrically_bounded_surface_shape_representation**, **geometrically_bounded_wireframe_shape_representation**, **location_shape_representation**, **manifold_surface_shape_representation**, **path_shape_representation**, **planar_shape_representation**, **shape_dimension_representation**, **shape_representation_with_parameters**, **shell_based_wireframe_shape_representation**

EXPRESS specification:

```
*)
 RULE subtype_exclusiveness_shape_representation FOR (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY(a <* shape_representation | NOT (type_check_function(a,[
     'PROCESS_PLANNING_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
     'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION',
     'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
     'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION',
     'PROCESS_PLANNING_SCHEMA.FACETED_BREP_SHAPE_REPRESENTATION',
'PROCESS_PLANNING_SCHEMA.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION',

'PROCESS_PLANNING_SCHEMA.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATI
ON',
     'PROCESS_PLANNING_SCHEMA.LOCATION_SHAPE_REPRESENTATION',
     'PROCESS_PLANNING_SCHEMA.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
```

 

```
    'PROCESS_PLANNING_SCHEMA.PATH_SHAPE_REPRESENTATION',
    'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
    'PROCESS_PLANNING_SCHEMA.SHAPE_DIMENSION_REPRESENTATION',
    'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
    'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION'],3
    )))) = 0;
 END_RULE;
(*
```

Argument definitions:

**shape_representation:** the set of all instances of **shape_representation** entities.

Formal propositions:

**WR1:** Each instance of the subtypes of **shape_representation** shall be **advanced_brep_shape_representation**, **direction_shape_representation**, **edge_based_wireframe_shape_representation**, **face_shape_representation**, **faceted_brep_shape_representation**, **geometrically_bounded_surface_shape_representation**, **geometrically_bounded_wireframe_shape_representation**, **location_shape_representation**, **manifold_surface_shape_representation**, **path_shape_representation**, **planar_shape_representation**, **shape_dimension_representation**, **shape_representation_with_parameters**, **shell_based_wireframe_shape_representation**

```
*)
END_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.

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: Six conformance classes are defined. Conformance to this part of ISO 10303 requires, as a minimum, conformance to class 1. Options are defined by classes 2 through 6 and may be selected by an implementation. Support for a particular conformance class requires support of all the options specified in that class.

Conformance to a particular class requires that all AIM elements defined as part of that class be supported.

Table 12 defines the classes to which each AIM element belongs.

The conformance classes are characterized as follows:

— Class 1: Class 6 and shapes represented by advanced b-rep.

— Class 2: Class 6 and shapes represented by non topological surface and wireframe models;

— Class 3: Class 6 and shapes represented by wireframe models with topology;

— Class 4: Class 6 and shapes represented by manifold surface models with topology;

— Class 5: Class 6 and shapes represented by faceted b-rep;

— Class 6: NC process plan information without shape;

Class 6 is a prerequisite for classes 1 through 5. If an implementation conforms to any of classes 1 through 5, then it shall also conform to class 6.

Conformance classes 1 through 5 are defined in terms of required AIM **shape_representation** subtypes.

## Table 1 — Conformance class elements

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Action | X | X | X | X | X | X |
| Action_assignment | X | X | X | X | X | X |
| Action_directive | X | X | X | X | X | X |
| Action_method | X | X | X | X | X | X |
| Action_method_assignment | X | X | X | X | X | X |
| Action_method_relationship | X | X | X | X | X | X |
| Action_method_role | X | X | X | X | X | X |
| Action_method_with_associated_documents | X | X | X | X | X | X |
| Action_property | X | X | X | X | X | X |
| Action_property_representation | X | X | X | X | X | X |
| Action_relationship | X | X | X | X | X | X |
| Action_request_assignment | X | X | X | X | X | X |
| Action_request_solution | X | X | X | X | X | X |
| Action_resource | X | X | X | X | X | X |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Action_resource_relationship | X | X | X | X | X | X |
| Action_resource_requirement | X | X | X | X | X | X |
| Action_resource_type | X | X | X | X | X | X |
| Address | X | X | X | X | X | X |
| Advanced_brep_shape_representation | X | | | | | |
| Advanced_face | X | | | X | | |
| Allowed_time | X | X | X | X | X | X |
| Alternate_action_method_relationship | X | X | X | X | X | X |
| Alternate_plan_relationship | X | X | X | X | X | X |
| Ancillary_activity | X | X | X | X | X | X |
| Ancillary_setup | X | X | X | X | X | X |
| Angular_location | X | X | X | X | X | |
| Angular_size | X | X | X | X | X | |
| Angularity_tolerance | X | X | X | X | X | |
| Apex | X | X | X | X | X | |
| Application_context | X | X | X | X | X | X |
| Application_context_element | X | X | X | X | X | X |
| Application_protocol_definition | X | X | X | X | X | X |
| Applied_action_assignment | X | X | X | X | X | X |
| Applied_action_method_assignment | X | X | X | X | X | X |
| Applied_action_request_assignment | X | X | X | X | X | X |
| Applied_approval_assignment | X | X | X | X | X | X |
| Applied_area | X | | | | | |
| Applied_classification_assignment | X | X | X | X | X | X |
| Applied_contract_assignment | X | X | X | X | X | X |
| Applied_date_assignment | X | X | X | X | X | X |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Applied_document_reference | X | X | X | X | X | X |
| Applied_document_usage_constraint_assignment | X | X | X | X | X | X |
| Applied_effectivity_assignment | X | X | X | X | X | X |
| Applied_external_identification_assignment | X | X | X | X | X | X |
| Applied_group_assignment | X | X | X | X | X | X |
| Applied_identification_assignment | X | X | X | X | X | X |
| Applied_library_assignment | X | X | X | X | X | X |
| Applied_organization_assignment | X | X | X | X | X | X |
| Applied_security_classification_assignment | X | X | X | X | X | X |
| Approval | X | X | X | X | X | X |
| Approval_assignment | X | X | X | X | X | X |
| Approval_date_time | X | X | X | X | X | X |
| Approval_person_organization | X | X | X | X | X | X |
| Approval_role | X | X | X | X | X | X |
| Approval_status | X | X | X | X | X | X |
| Assembly_component_usage | X | X | X | X | X | X |
| Axis1_placement | X | X | | X | | |
| Axis2_placement_2d | X | X | | X | | |
| Axis2_placement_3d | X | X | X | X | X | |
| B_spline_curve | X | X | X | X | | |
| B_spline_curve_with_knots | X | X | X | X | | |
| B_spline_surface | X | X | | X | | |
| B_spline_surface_with_knots | X | X | | X | | |
| Bezier_curve | X | X | X | X | | |
| Bezier_surface | X | X | | X | | |
| Block_shape_representation | X | X | X | X | X | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Boss | X | | | | | |
| Boss_top | X | | | | | |
| Boundary_curve | X | X | | | | |
| Bounded_curve | X | X | X | X | | |
| Bounded_pcurve | X | | | | | |
| Bounded_surface | X | X | | X | | |
| Bounded_surface_curve | X | X | | | | |
| Brep_with_voids | X | | | | X | |
| Calendar_date | X | X | X | X | X | X |
| Cartesian_point | X | X | X | X | X | |
| Cartesian_transformation_operator | X | X | X | X | X | |
| Cartesian_transformation_operator_3d | X | X | X | X | X | |
| Centre_of_symmetry | X | X | X | X | X | |
| Chamfer | X | | | | | |
| Chamfer_offset | X | | | | | |
| Characterized_object | X | X | X | X | X | X |
| Circle | X | X | X | X | | |
| Circular_closed_profile | X | | | | | |
| Circular_pattern | X | | | | | |
| Circular_runout_tolerance | X | X | X | X | X | |
| Class | X | X | X | X | X | X |
| Classification_assignment | X | X | X | X | X | X |
| Classification_role | X | X | X | X | X | X |
| Closed_path_profile | X | | | | | |
| Closed_shell | X | | | X | X | |
| Coaxiality_tolerance | X | X | X | X | X | |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Common_datum | X | X | X | X | X | |
| Composite_curve | X | X | | | | |
| Composite_curve_on_surface | X | X | | | | |
| Composite_curve_segment | X | X | | | | |
| Composite_hole | X | | | | | |
| Composite_shape_aspect | X | X | X | X | X | |
| Compound_feature | X | | | | | |
| Compound_representation_item | X | X | X | X | X | X |
| Concentricity_tolerance | X | X | X | X | X | |
| Conic | X | X | X | X | | |
| Conical_surface | | X | | | | |
| Connected_edge_set | | | X | | | |
| Connected_face_set | X | | | X | | |
| Context_dependent_unit | X | X | X | X | X | X |
| Continuous_process_relationship | X | X | X | X | X | X |
| Contract | X | X | X | X | X | X |
| Contract_assignment | X | X | X | X | X | X |
| Contract_type | X | X | X | X | X | X |
| Controller | X | X | X | X | X | X |
| Controller_program | X | X | X | X | X | X |
| Conversion_based_unit | X | X | X | X | X | X |
| Coordinated_universal_time_offset | X | X | X | X | X | X |
| Curve | X | X | X | X | | |
| Curve_bounded_surface | X | X | | | | |
| Curve_replica | X | X | X | X | | |
| Cylindrical_shape_representation | X | X | X | X | X | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Cylindrical_surface | X | X | | | | |
| Cylindricity_tolerance | X | X | | X | X | |
| Data_environment | X | X | X | X | X | X |
| Date | X | X | X | X | X | X |
| Date_and_time | X | X | X | X | X | X |
| Date_assignment | X | X | X | X | X | X |
| Date_role | X | X | X | X | X | X |
| Datum | X | X | X | X | X | |
| Datum_feature | X | X | X | X | X | |
| Datum_reference | X | X | X | X | X | |
| Datum_target | X | X | X | X | X | |
| Defining_action_method_relationship | X | X | X | X | X | X |
| Definitional_representation | X | X | X | X | X | X |
| Degenerate_pcurve | | X | | X | | |
| Degenerate_toroidal_surface | | X | | X | | |
| Derived_shape_aspect | X | X | X | X | X | X |
| Derived_unit | X | X | X | X | X | X |
| Derived_unit_element | X | X | X | X | X | X |
| Description_attribute | X | X | X | X | X | X |
| Descriptive_representation_item | X | X | X | X | X | X |
| Design_reference | X | X | X | X | X | X |
| Dimension_related_tolerance_zone_element | X | X | X | X | X | |
| Dimensional_characteristic_representation | X | X | X | X | X | |
| Dimensional_exponents | X | X | X | X | X | X |
| Dimensional_location | X | X | X | X | X | |
| Dimensional_location_with_path | X | X | X | X | X | |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Dimensional_size | X | X | X | X | X | |
| Dimensional_size_with_path | X | X | X | X | X | |
| Directed_action | X | X | X | X | X | X |
| Directed_dimensional_location | X | X | X | X | X | X |
| Direction | X | X | X | X | X | |
| Direction_shape_representation | X | X | X | X | X | |
| Document | X | X | X | X | X | X |
| Document_file | X | X | X | X | X | X |
| Document_file_properties | X | X | X | X | X | X |
| Document_product_association | X | X | X | X | X | X |
| Document_reference | X | X | X | X | X | X |
| Document_relationship | X | X | X | X | X | X |
| Document_representation_type | X | X | X | X | X | X |
| Document_type | X | X | X | X | X | X |
| Document_usage_constraint | X | X | X | X | X | X |
| Document_usage_constraint_assignment | X | X | X | X | X | X |
| Document_usage_role | X | X | X | X | X | X |
| Edge | X | | X | X | X | |
| Edge_based_wireframe_model | | | X | | | |
| Edge_based_wireframe_shape_representation | | | X | | | |
| Edge_curve | X | | X | X | | |
| Edge_loop | X | | X | X | | |
| Edge_round | X | | | | | |
| Effectivity | X | X | X | X | X | X |
| Effectivity_assignment | X | X | X | X | X | X |
| Elementary_surface | | | X | | X | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Ellipse | X | X | X | X | | |
| Evaluated_degenerate_pcurve | X | X | X | X | | |
| Executed_action | X | X | X | X | X | X |
| Extension | X | X | X | X | X | X |
| External_identification_assignment | X | X | X | X | X | X |
| External_source | X | X | X | X | X | X |
| Externally_defined_class | X | X | X | X | X | X |
| Externally_defined_dimension_definition | X | X | X | X | X | |
| Externally_defined_feature_definition | X | X | X | X | X | |
| Externally_defined_general_property | X | X | X | X | X | X |
| Externally_defined_item | X | X | X | X | X | X |
| Externally_defined_item_relationship | X | X | X | X | X | X |
| Externally_defined_representation_with_parameters | X | X | X | X | X | X |
| Externally_defined_schema | X | X | X | X | X | X |
| Face | X | | | X | X | |
| Face_bound | X | | | X | X | |
| Face_outer_bound | X | | | X | X | |
| Face_shape_representation | X | | | | | |
| Face_surface | X | | | X | X | |
| Faceted_brep | | | | | X | |
| Faceted_brep_shape_representation | | | | | X | |
| Feature_component_definition | X | | | | | |
| Feature_component_relationship | X | | | | | |
| Feature_definition | X | | | | | |
| Feature_dependency | X | | | | | |
| Feature_identification_item | X | | | | | |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Feature_interaction | X | | | | | |
| Feature_pattern | X | | | | | |
| Feature_process | X | X | X | X | X | X |
| Fillet | X | | | | | |
| Fixture_assembly | X | X | X | X | X | X |
| Fixture_assembly_element | X | X | X | X | X | X |
| Fixture_assembly_relationship | X | X | X | X | X | X |
| Fixture_contract | X | X | X | X | X | X |
| Fixture_machine_mounting | X | X | X | X | X | X |
| Fixture_machine_unmounting | X | X | X | X | X | X |
| Fixture_pallet_mounting | X | X | X | X | X | X |
| Fixture_pallet_unmounting | X | X | X | X | X | X |
| Fixture_setup | X | X | X | X | X | X |
| Flat_face | X | | | | | |
| Flatness_tolerance | X | X | X | X | X | |
| Founded_item | X | X | X | X | X | |
| Functionally_defined_transformation | X | X | X | X | X | |
| Gear | X | | | | | |
| General_property | X | X | X | X | X | X |
| General_property_association | X | X | X | X | X | X |
| Generic_manufacturing_resource | X | X | X | X | X | X |
| Geometric_alignment | X | X | X | X | X | |
| Geometric_curve_set | | X | | | | |
| Geometric_intersection | X | X | X | X | | |
| Geometric_representation_context | X | X | X | X | X | |
| Geometric_representation_item | X | X | X | X | X | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Geometric_set | | X | | | | |
| Geometric_tolerance | X | X | X | X | X | |
| Geometric_tolerance_relationship | X | X | X | X | X | |
| Geometric_tolerance_with_datum_reference | X | X | X | X | X | |
| Geometric_tolerance_with_defined_unit | X | X | X | X | X | |
| Geometrically_bounded_surface_shape_representation | | X | | | | |
| Geometrically_bounded_wireframe_shape_representation | | X | | | | |
| Global_uncertainty_assigned_context | X | X | X | X | X | X |
| Global_unit_assigned_context | X | X | X | X | X | X |
| Group | X | X | X | X | X | X |
| Group_assignment | X | X | X | X | X | X |
| Group_relationship | X | X | X | X | X | X |
| Hole_bottom | X | | | | | |
| Hyperbola | X | X | X | X | X | |
| Id_attribute | X | X | X | X | X | X |
| Identification_assignment | X | X | X | X | X | X |
| Identification_role | X | X | X | X | X | X |
| Illustration | X | X | X | X | X | X |
| In_facility_location | X | X | X | X | X | X |
| Instanced_feature | X | | | | | |
| Intersection_curve | | X | | X | | |
| Known_source | X | X | X | X | X | X |
| Length_measure_with_unit | X | X | X | X | X | X |
| Length_unit | X | X | X | X | X | X |
| Library_class_version_assignment | X | X | X | X | X | X |
| Library_property_version_assignment | X | X | X | X | X | X |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Limits_and_fits | X | X | X | X | X | |
| Line | X | X | X | X | | |
| Line_profile_tolerance | X | X | X | X | X | |
| Linear_profile | X | | | | | |
| Local_time | X | X | X | X | X | X |
| Location_shape_representation | X | X | X | X | X | |
| Loop | X | | X | X | X | |
| Machine | X | X | X | X | X | X |
| Machine_element_relationship | X | X | X | X | X | X |
| Machine_setup | X | X | X | X | X | X |
| Machine_usage | X | X | X | X | X | X |
| Machining_operation | X | X | X | X | X | X |
| Machining_process | X | X | X | X | X | X |
| Machining_tolerance | X | X | X | X | X | X |
| Make_from_usage_option | X | X | X | X | X | X |
| Manifold_solid_brep | X | | | | X | |
| Manifold_surface_shape_representation | | | | X | | |
| Manufacturing_activity | X | X | X | X | X | X |
| Manufacturing_process | X | X | X | X | X | X |
| Manufacturing_process_relationship | X | X | X | X | X | X |
| Mapped_item | X | X | X | X | X | X |
| Marking | X | | | | | |
| Mass_measure_with_unit | X | X | X | X | X | X |
| Mass_unit | X | X | X | X | X | X |
| Material_designation | X | X | X | X | X | X |
| Material_property | X | X | X | X | X | X |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Material_property_representation | X | X | X | X | X | X |
| Measure_representation_item | X | X | X | X | X | X |
| Measure_qualification | X | X | X | X | X | X |
| Measure_with_unit | X | X | X | X | X | X |
| Modified_geometric_tolerance | X | X | X | X | X | |
| Modified_pattern | X | | | | | |
| Name_attribute | X | X | X | X | X | X |
| Named_unit | X | X | X | X | X | X |
| Next_assembly_usage_occurrence | X | X | X | X | X | X |
| Ngon_closed_profile | X | | | | | |
| Ngon_shape_representation | X | X | X | X | X | X |
| Non_machining_process | X | X | X | X | X | X |
| Non_machining_process_relationship | X | X | X | X | X | X |
| Object_role | X | X | X | X | X | X |
| Offset_curve_3d | | X | X | X | | |
| Offset_surface | | X | | X | | |
| Open_path_profile | X | | | | | |
| Open_shell | X | | | X | X | |
| Organization | X | X | X | X | X | X |
| Organization_assignment | X | X | X | X | X | X |
| Organization_role | X | X | X | X | X | X |
| Organizational_address | X | X | X | X | X | X |
| Organizational_project | X | X | X | X | X | X |
| Oriented_closed_shell | X | | | X | X | |
| Oriented_edge | X | | X | X | X | |
| Oriented_face | X | | | X | X | |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Oriented_open_shell | X | | | X | X | |
| Oriented_path | X | | | X | X | |
| Oriented_surface | X | | | | X | |
| Outer_boundary_curve | | X | | | | |
| Outer_round | X | | | | | |
| Outside_profile | X | | | | | |
| Pallet | X | X | X | X | X | X |
| Pallet_machine_mounting | X | X | X | X | X | X |
| Parabola | X | X | X | X | X | |
| Parallel_offset | X | X | X | X | X | |
| Parallelism_tolerance | X | X | X | X | X | |
| Parametric_representation_context | X | X | X | X | X | X |
| Part_contract | X | X | X | X | X | X |
| Part_fixture_mounting | X | X | X | X | X | X |
| Part_fixture_unmounting | X | X | X | X | X | X |
| Part_routing | X | X | X | X | X | X |
| Part_mounting | X | X | X | X | X | X |
| Part_machine_mounting | X | X | X | X | X | X |
| Part_machine_unmounting | X | X | X | X | X | X |
| Partial_circular_profile | X | | | | | |
| Path | X | | X | X | X | |
| Path_feature_component | X | | | | | |
| Path_shape_representation | X | | | | | |
| Pattern_offset_membership | X | | | | | |
| Pattern_omit_membership | X | | | | | |
| Pcurve | | X | | X | | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Perpendicular_to | X | X | X | X | X | |
| Perpendicularity_tolerance | X | X | X | X | X | |
| Person | X | X | X | X | X | X |
| Person_and_organization | X | X | X | X | X | X |
| Personal_address | X | X | X | X | X | X |
| Placement | X | X | X | X | X | |
| Planar_shape_representation | X | | | | | |
| Plane | X | X | | X | X | |
| Plane_angle_measure_with_unit | X | X | X | X | X | X |
| Plane_angle_unit | X | X | X | X | X | X |
| Plus_minus_tolerance | X | X | X | X | X | |
| Pocket | X | | | | | |
| Pocket_bottom | X | | | | | |
| Point | X | X | X | X | X | |
| Point_on_curve | | X | X | X | X | |
| Point_on_surface | | X | | X | X | |
| Point_replica | | X | X | | | |
| Poly_loop | X | | | X | X | |
| Polyline | X | X | X | X | | |
| Position_tolerance | X | X | X | X | X | |
| Pre_defined_item | X | X | X | X | X | X |
| Precision_qualifier | X | X | X | X | X | X |
| Process_activity | X | X | X | X | X | X |
| Process_plan_activity | X | X | X | X | X | X |
| Process_plan_security | X | X | X | X | X | X |
| Process_plan_specification | X | X | X | X | X | X |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Process_plan_version | X | X | X | X | X | X |
| Process_product_association | X | X | X | X | X | X |
| Process_property_association | X | X | X | X | X | X |
| Product | X | X | X | X | X | X |
| Product_category | X | X | X | X | X | X |
| Product_context | X | X | X | X | X | X |
| Product_definition | X | X | X | X | X | X |
| Product_definition_context | X | X | X | X | X | X |
| Product_definition_effectivity | X | X | X | X | X | X |
| Product_definition_formation | X | X | X | X | X | X |
| Product_definition_process | X | X | X | X | X | X |
| Product_definition_relationship | X | X | X | X | X | X |
| Product_definition_shape | X | X | X | X | X | X |
| Product_definition_usage | X | X | X | X | X | X |
| Product_definition_with_associated_documents | X | X | X | X | X | X |
| Product_related_product_category | X | X | X | X | X | X |
| Production_rate | X | X | X | X | X | X |
| Profile_floor | X | | | | | |
| Projected_zone_definition | X | X | X | X | X | |
| Property_definition | X | X | X | X | X | X |
| Property_definition_relationship | X | X | X | X | X | X |
| Property_definition_representation | X | X | X | X | X | X |
| Property_process | X | X | X | X | X | X |
| Protrusion | X | | | | | |
| Qualified_representation_item | X | X | X | X | X | X |
| Quasi_uniform_curve | X | X | X | X | | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Quasi_uniform_surface | X | X | | X | | |
| Range_of_parts | X | X | X | X | X | X |
| Ratio_measure_with_unit | X | X | X | X | X | X |
| Ratio_unit | X | X | X | X | X | X |
| Rational_b_spline_curve | X | X | X | X | | |
| Rational_b_spline_surface | X | X | | X | | |
| Rectangular_closed_profile | X | | | | | |
| Rectangular_composite_surface | | X | | | | |
| Rectangular_pattern | X | | | | | |
| Rectangular_trimmed_surface | | X | | | | |
| Referenced_modified_datum | X | X | X | X | X | |
| Relationship_condition | X | X | X | X | X | X |
| Removal_volume | X | | | | | |
| Reparametrised_composite_curve_segment | | X | | | | |
| Replicate_feature | X | | | | | |
| Representation | X | X | X | X | X | X |
| Representation_context | X | X | X | X | X | X |
| Representation_item | X | X | X | X | X | X |
| Representation_map | X | X | X | X | X | X |
| Requirement_for_action_resource | X | X | X | X | X | X |
| Resource_property | X | X | X | X | X | X |
| Resource_property_representation | X | X | X | X | X | X |
| Resource_requirement_type | X | X | X | X | X | X |
| Resource_with_material | X | X | X | X | X | X |
| Resource_with_representation | X | X | X | X | X | X |
| Revision | X | X | X | X | X | X |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Revolved_profile | X | | | | | |
| Rib_top | X | | | | | |
| Rib_top_floor | X | | | | | |
| Role_association | X | X | X | X | X | X |
| Round_hole | X | | | | | |
| Rounded_end | X | | | | | |
| Rounded_u_profile | X | | | | | |
| Roundness_tolerance | X | X | X | X | X | |
| Runout_zone_definition | X | X | X | X | X | |
| Runout_zone_orientation | X | X | X | X | X | |
| Runout_zone_orientation_reference_direction | X | X | X | X | X | |
| Seam_curve | | X | | X | | |
| Security_classification | X | X | X | X | X | X |
| Security_classification_assignment | X | X | X | X | X | X |
| Security_classification_level | X | X | X | X | X | X |
| Sequential_method | X | X | X | X | X | X |
| Serial_action_method | X | X | X | X | X | X |
| Shape_aspect | X | X | X | X | X | X |
| Shape_aspect_deriving_relationship | X | X | X | X | X | X |
| Shape_aspect_relationship | X | X | X | X | X | X |
| Shape_defining_relationship | X | X | X | X | X | |
| Shape_definition_representation | X | X | X | X | X | |
| Shape_dimension_representation | X | X | X | X | X | |
| Shape_representation | X | X | X | X | X | |
| Shape_representation_with_parameters | X | | | | | |
| Shell_based_surface_model | | | | X | | |

 

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Shell_based_wireframe_model | | | X | | | |
| Shell_based_wireframe_shape_representation | | | X | | | |
| Si_unit | X | X | X | X | X | X |
| Single_activity_relationship | X | X | X | X | X | X |
| Slot | X | | | | | |
| Slot_end | X | | | | | |
| Solid_angle_unit | X | X | X | X | X | X |
| Solid_model | X | | | | X | |
| Spherical_cap | X | | | | | |
| Spherical_surface | X | X | | X | | |
| Square_u_profile | X | | | | | |
| Standard_uncertainty | X | X | X | X | X | X |
| Step | X | | | | | |
| Straightness_tolerance | X | X | X | X | X | |
| Surface | X | X | | X | X | |
| Surface_curve | | X | | X | | |
| Surface_of_linear_extrusion | X | X | | X | | |
| Surface_of_revolution | X | X | | X | | |
| Surface_patch | | X | | | | |
| Surface_profile_tolerance | X | X | X | X | X | |
| Surface_replica | | X | | X | | |
| Swept_surface | X | X | | X | | |
| Symmetric_shape_aspect | X | X | X | X | X | |
| Symmetry_tolerance | X | X | X | X | X | |
| Tangent | X | X | X | X | X | |
| Taper | X | | | | | |

## Table 1 — Conformance class elements (continued)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Tee_profile | X | | | | | |
| Thread | X | | | | | |
| Time_measure_with_unit | X | X | X | X | X | X |
| Time_unit | X | X | X | X | X | X |
| Tolerance_value | X | X | X | X | X | |
| Tolerance_zone | X | X | X | X | X | |
| Tolerance_zone_definition | X | X | X | X | X | |
| Tolerance_zone_form | X | X | X | X | X | |
| Tool_assembly | X | X | X | X | X | X |
| Tool_assembly_element | X | X | X | X | X | X |
| Tool_assembly_relationship | X | X | X | X | X | X |
| Tool_body | X | X | X | X | X | X |
| Tool_contract | X | X | X | X | X | X |
| Tool_placement_instruction | X | X | X | X | X | X |
| Tool_magazine_turret_carousel | X | X | X | X | X | X |
| Tool_setup | X | X | X | X | X | X |
| Topological_representation_item | X | | X | X | X | |
| Toroidal_surface | | X | | | | |
| Total_runout_tolerance | X | X | X | X | X | |
| Transition_feature | X | | | | | |
| Trimmed_curve | | X | | | | |
| Turned_knurl | X | | | | | |
| Type_qualifier | X | X | X | X | X | X |
| Uncertainty_measure_with_unit | X | X | X | X | X | X |
| Uncertainty_qualifier | X | X | X | X | X | X |
| Uniform_curve | X | X | X | X | | |

## Table 1 — Conformance class elements (concluded)

| Aim Element | class | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Uniform_surface | X | X | | X | | |
| Validation | X | X | X | X | X | X |
| Value_range | X | X | X | X | X | X |
| Value_representation_item | X | X | X | X | X | X |
| Vector | X | X | X | X | X | |
| Vee_profile | X | | | | | |
| Versioned_action_request | X | X | X | X | X | X |
| Vertex | X | | X | X | X | |
| Vertex_loop | X | | X | X | X | |
| Vertex_point | X | | X | X | X | |
| Vertex_shell | X | | X | X | X | |
| View_reference | X | X | X | X | X | X |
| Wire_shell | | | X | | | |
| Work_cell | X | X | X | X | X | X |
| Workstation | X | X | X | X | X | X |

# Annex A
(normative)

# AIM EXPRESS expanded listing

```
SCHEMA process_planning_schema;

 CONSTANT
 dummy_gri : geometric_representation_item := representation_item('') ||
         geometric_representation_item();
 dummy_tri : topological_representation_item := representation_item('')
         || topological_representation_item();
 END_CONSTANT;

 TYPE action_item = SELECT
  (feature_dependency,
   product_definition_formation);
 END_TYPE; -- action_item

 TYPE action_method_item = SELECT
  (externally_defined_schema);
 END_TYPE; -- action_method_item

 TYPE action_request_item = SELECT
  (product_definition_formation,
   fixture_contract,
   tool_contract);
 END_TYPE; -- action_request_item

 TYPE ahead_or_behind = ENUMERATION OF
  (exact,
   ahead,
   behind);
 END_TYPE; -- ahead_or_behind

 TYPE amount_of_substance_measure = REAL;
 END_TYPE; -- amount_of_substance_measure

 TYPE angle_relator = ENUMERATION OF
  (equal,
   small,
   large);
 END_TYPE; -- angle_relator

 TYPE approval_item = SELECT
  (revision);
 END_TYPE; -- approval_item

 TYPE area_measure = REAL;
 END_TYPE; -- area_measure

 TYPE attribute_type = SELECT
  (label,
   text);
 END_TYPE; -- attribute_type

 TYPE axis2_placement = SELECT
```

 

```
 (axis2_placement_2d,
  axis2_placement_3d);
END_TYPE; -- axis2_placement

TYPE b_spline_curve_form = ENUMERATION OF
 (elliptic_arc,
  polyline_form,
  parabolic_arc,
  circular_arc,
  unspecified,
  hyperbolic_arc);
END_TYPE; -- b_spline_curve_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

TYPE boolean_operand = SELECT
 (solid_model);
END_TYPE; -- boolean_operand

TYPE celsius_temperature_measure = REAL;
END_TYPE; -- celsius_temperature_measure

TYPE characterized_action_definition = SELECT
 (action,
  action_method,
  action_method_relationship,
  action_relationship);
END_TYPE; -- characterized_action_definition

TYPE characterized_definition = SELECT
 (characterized_object,
  characterized_product_definition,
  shape_definition);
END_TYPE; -- characterized_definition

TYPE characterized_material_property = SELECT
 (material_property_representation);
END_TYPE; -- characterized_material_property

TYPE characterized_product_definition = SELECT
 (product_definition,
  product_definition_relationship);
END_TYPE; -- characterized_product_definition

TYPE characterized_resource_definition = SELECT
 (action_resource,
  action_resource_relationship,
  action_resource_requirement);
END_TYPE; -- characterized_resource_definition
```

```
TYPE classification_item = SELECT
 (externally_defined_representation_with_parameters);
END_TYPE; -- classification_item

TYPE compound_item_definition = SELECT
 (list_representation_item,
  set_representation_item);
END_TYPE; -- compound_item_definition

TYPE contract_item = SELECT
 (process_plan_specification,
  document_file,
  fixture_assembly,
  product_definition_formation,
  tool_assembly,
  in_facility_location);
END_TYPE; -- contract_item

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure

TYPE curve_on_surface = SELECT
 (pcurve,
  surface_curve,
  composite_curve_on_surface);
END_TYPE; -- curve_on_surface

TYPE date_item = SELECT
 (process_plan_security,
  versioned_action_request);
END_TYPE; -- date_item

TYPE date_time_or_event_occurrence = SELECT
 (date_time_select);
END_TYPE; -- date_time_or_event_occurrence

TYPE date_time_select = SELECT
 (date,
  local_time,
  date_and_time);
END_TYPE; -- date_time_select

TYPE day_in_month_number = INTEGER;
WHERE
 wr1: (1 <= SELF) AND (SELF <= 31);
END_TYPE; -- day_in_month_number

TYPE derived_property_select = SELECT
 (property_definition,
  action_property,
  resource_property);
END_TYPE; -- derived_property_select

TYPE description_attribute_select = SELECT
 (action_request_solution,
  application_context,
  approval_role,
  date_role,
  effectivity,
  external_source,
  organization_role,
```

 

```
  person_and_organization,
  property_definition_representation,
  representation);
END_TYPE; -- description_attribute_select

TYPE dimension_count = INTEGER;
WHERE
 wr1: SELF > 0;
END_TYPE; -- dimension_count

TYPE dimensional_characteristic = SELECT
 (dimensional_location,
  dimensional_size);
END_TYPE; -- dimensional_characteristic

TYPE document_reference_item = SELECT
 (action_method,
  action_resource_requirement,
  controller,
  date_and_time,
  externally_defined_schema,
  externally_defined_dimension_definition,
  fixture_assembly,
  fixture_assembly_element,
  machine_usage,
  manufacturing_activity,
  process_plan_activity,
  product_definition,
  product_definition_formation,
  property_definition,
  manufacturing_process,
  tool_assembly,
  tool_assembly_element,
  tool_placement_instruction,
  workstation);
END_TYPE; -- document_reference_item

TYPE document_usage_constraint_item = SELECT
 (illustration);
END_TYPE; -- document_usage_constraint_item

TYPE effectivity_item = SELECT
 (product_definition_formation);
END_TYPE; -- effectivity_item

TYPE electric_current_measure = REAL;
END_TYPE; -- electric_current_measure

TYPE external_identification_item = SELECT
 (document,
  externally_defined_class,
  externally_defined_general_property);
END_TYPE; -- external_identification_item

TYPE founded_item_select = SELECT
 (founded_item,
  representation_item);
END_TYPE; -- founded_item_select

TYPE geometric_set_select = SELECT
 (point,
```

```
  curve,
  surface);
END_TYPE; -- geometric_set_select

TYPE group_item = SELECT
 (workstation,
  instanced_feature,
  feature_process);
END_TYPE; -- group_item

TYPE hour_in_day = INTEGER;
WHERE
 wr1: (0 <= SELF) AND (SELF < 24);
END_TYPE; -- hour_in_day

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

TYPE identification_assignment_item = SELECT
 (document_file);
END_TYPE; -- identification_assignment_item

TYPE identifier = STRING;
END_TYPE; -- identifier

TYPE knot_type = ENUMERATION OF
 (uniform_knots,
  quasi_uniform_knots,
  piecewise_bezier_knots,
  unspecified);
END_TYPE; -- knot_type

TYPE label = STRING;
END_TYPE; -- label

TYPE length_measure = REAL;
END_TYPE; -- length_measure

TYPE limit_condition = ENUMERATION OF
 (regardless_of_feature_size,
  maximum_material_condition,
  least_material_condition);
END_TYPE; -- limit_condition

TYPE list_of_reversible_topology_item = LIST [0:?] OF
      reversible_topology_item;
END_TYPE; -- list_of_reversible_topology_item

TYPE list_representation_item = LIST [1:?] OF representation_item;
END_TYPE; -- list_representation_item
```

 

```
TYPE luminous_intensity_measure = REAL;
END_TYPE; -- luminous_intensity_measure

TYPE mass_measure = REAL;
END_TYPE; -- mass_measure

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

TYPE minute_in_hour = INTEGER;
WHERE
 wr1: (0 <= SELF) AND (SELF <= 59);
END_TYPE; -- minute_in_hour

TYPE month_in_year_number = INTEGER;
WHERE
 wr1: (1 <= SELF) AND (SELF <= 12);
END_TYPE; -- month_in_year_number

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

TYPE organization_item = SELECT
 (process_plan_activity,
  fixture_assembly_element,
  known_source,
  machine,
  part_contract,
  product_definition,
  tool_assembly_element);
END_TYPE; -- organization_item

TYPE parameter_value = REAL;
END_TYPE; -- parameter_value

TYPE pcurve_or_surface = SELECT
 (pcurve,
```

```
  surface);
END_TYPE; -- pcurve_or_surface

TYPE person_organization_select = SELECT
 (person,
  organization,
  person_and_organization);
END_TYPE; -- person_organization_select

TYPE plane_angle_measure = REAL;
END_TYPE; -- plane_angle_measure

TYPE positive_length_measure = length_measure;
WHERE
 wr1: SELF > 0;
END_TYPE; -- positive_length_measure

TYPE positive_plane_angle_measure = plane_angle_measure;
WHERE
 wr1: SELF > 0;
END_TYPE; -- positive_plane_angle_measure

TYPE preferred_surface_curve_representation = ENUMERATION OF
 (pcurve_s2,
  pcurve_s1,
  curve_3d);
END_TYPE; -- preferred_surface_curve_representation

TYPE process_or_process_relationship = SELECT
 (product_definition_process,
  property_process,
  relationship_with_condition);
END_TYPE; -- process_or_process_relationship

TYPE product_or_formation_or_definition = SELECT
 (product,
  product_definition_formation,
  product_definition);
END_TYPE; -- product_or_formation_or_definition

TYPE property_or_shape_select = SELECT
 (property_definition,
  shape_definition);
END_TYPE; -- property_or_shape_select

TYPE ratio_measure = REAL;
END_TYPE; -- ratio_measure

TYPE relationship_with_condition = SELECT
 (action_method_relationship,
  action_relationship);
END_TYPE; -- relationship_with_condition

TYPE represented_definition = SELECT
 (general_property,
  property_definition,
  property_definition_relationship,
  shape_aspect,
  shape_aspect_relationship);
END_TYPE; -- represented_definition
```

 

```
TYPE reversible_topology = SELECT
 (reversible_topology_item,
  list_of_reversible_topology_item,
  set_of_reversible_topology_item);
END_TYPE; -- reversible_topology

TYPE reversible_topology_item = SELECT
 (edge,
  path,
  face,
  face_bound,
  closed_shell,
  open_shell);
END_TYPE; -- reversible_topology_item

TYPE role_select = SELECT
 (action_assignment,
  action_request_assignment,
  approval_assignment,
  approval_date_time,
  contract_assignment,
  document_reference,
  effectivity_assignment,
  group_assignment,
  security_classification_assignment);
END_TYPE; -- role_select

TYPE second_in_minute = REAL;
WHERE
 wr1: (0 <= SELF) AND (SELF <= 60);
END_TYPE; -- second_in_minute

TYPE security_classification_item = SELECT
 (process_plan_activity,
  process_plan_version);
END_TYPE; -- security_classification_item

TYPE set_of_reversible_topology_item = SET [0:?] OF
      reversible_topology_item;
END_TYPE; -- set_of_reversible_topology_item

TYPE set_representation_item = SET [1:?] OF representation_item;
END_TYPE; -- set_representation_item

TYPE shape_definition = SELECT
 (product_definition_shape,
  shape_aspect,
  shape_aspect_relationship);
END_TYPE; -- shape_definition

TYPE shell = SELECT
 (vertex_shell,
  wire_shell,
  open_shell,
  closed_shell);
END_TYPE; -- shell

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

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

TYPE solid_angle_measure = REAL;
END_TYPE; -- solid_angle_measure

TYPE source_item = SELECT
 (identifier);
END_TYPE; -- source_item

TYPE supported_item = SELECT
 (action_directive,
  action,
  action_method);
END_TYPE; -- supported_item

TYPE surface_boundary = SELECT
 (boundary_curve,
```

 

```
  degenerate_pcurve);
END_TYPE; -- surface_boundary

TYPE surface_model = SELECT
 (shell_based_surface_model);
END_TYPE; -- surface_model

TYPE text = STRING;
END_TYPE; -- text

TYPE thermodynamic_temperature_measure = REAL;
END_TYPE; -- thermodynamic_temperature_measure

TYPE time_measure = REAL;
END_TYPE; -- time_measure

TYPE tolerance_method_definition = SELECT
 (tolerance_value,
  limits_and_fits);
END_TYPE; -- tolerance_method_definition

TYPE tolerance_select = SELECT
 (geometric_tolerance,
  plus_minus_tolerance);
END_TYPE; -- tolerance_select

TYPE transformation = SELECT
 (functionally_defined_transformation);
END_TYPE; -- transformation

TYPE transition_code = ENUMERATION OF
 (discontinuous,
  cont_same_gradient_same_curvature,
  cont_same_gradient,
  continuous);
END_TYPE; -- transition_code

TYPE trimming_preference = ENUMERATION OF
 (parameter,
  unspecified,
  cartesian);
END_TYPE; -- trimming_preference

TYPE trimming_select = SELECT
 (cartesian_point,
  parameter_value);
END_TYPE; -- trimming_select

TYPE unit = SELECT
 (named_unit,
  derived_unit);
END_TYPE; -- unit

TYPE value_qualifier = SELECT
 (precision_qualifier,
  type_qualifier,
  uncertainty_qualifier);
END_TYPE; -- value_qualifier

TYPE vector_or_direction = SELECT
 (vector,
```

```
  direction);
END_TYPE; -- vector_or_direction

TYPE volume_measure = REAL;
END_TYPE; -- volume_measure

TYPE wireframe_model = SELECT
 (shell_based_wireframe_model,
  edge_based_wireframe_model);
END_TYPE; -- wireframe_model

TYPE year_number = INTEGER;
END_TYPE; -- year_number

ENTITY action;
  name     : label;
  description  : OPTIONAL text;
  chosen_method : action_method;
 DERIVE
  id : identifier := get_id_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- action

ENTITY action_assignment
 ABSTRACT SUPERTYPE;
  assigned_action : action;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_assignment

ENTITY action_directive;
  name    : label;
  description : OPTIONAL text;
  analysis  : text;
  comment   : text;
  requests  : SET [1:?] OF versioned_action_request;
END_ENTITY; -- action_directive

ENTITY action_method;
  name    : label;
  description : OPTIONAL text;
  consequence : text;
  purpose   : text;
END_ENTITY; -- action_method

ENTITY action_method_assignment
 ABSTRACT SUPERTYPE;
  assigned_action_method : action_method;
  role          : action_method_role;
END_ENTITY; -- action_method_assignment

ENTITY action_method_relationship;
  name      : label;
  description   : OPTIONAL text;
  relating_method : action_method;
  related_method : action_method;
```

 

```
END_ENTITY; -- action_method_relationship

ENTITY action_method_role;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- action_method_role

ENTITY action_method_with_associated_documents
 SUBTYPE OF (action_method);
  documents : SET [1:?] OF document;
END_ENTITY; -- action_method_with_associated_documents

ENTITY action_property;
  name    : label;
  description : text;
  definition : characterized_action_definition;
END_ENTITY; -- action_property

ENTITY action_property_representation;
  name      : label;
  description  : text;
  property    : action_property;
  representation : representation;
END_ENTITY; -- action_property_representation

ENTITY action_relationship;
  name      : label;
  description   : OPTIONAL text;
  relating_action : action;
  related_action : action;
END_ENTITY; -- action_relationship

ENTITY action_request_assignment
 ABSTRACT SUPERTYPE;
  assigned_action_request : versioned_action_request;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- action_request_assignment

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,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- action_request_solution

ENTITY action_resource;
  name    : label;
  description : OPTIONAL text;
  usage    : SET [1:?] OF supported_item;
  kind    : action_resource_type;
END_ENTITY; -- action_resource
```

```
ENTITY action_resource_relationship;
  name       : label;
  description    : OPTIONAL text;
  relating_resource : action_resource;
  related_resource : action_resource;
END_ENTITY; -- action_resource_relationship

ENTITY action_resource_requirement;
  name    : label;
  description : text;
  kind    : resource_requirement_type;
  OPERATIONS : SET [1:?] OF characterized_action_definition;
END_ENTITY; -- action_resource_requirement

ENTITY action_resource_type;
  name : label;
END_ENTITY; -- action_resource_type

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

ENTITY advanced_brep_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MANIFOLD_SOLID_BREP',
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MANIFOLD_SOLID_BREP',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       > 0;
  wr3: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_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 ('PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fcs))) ))
       = 0)) )) = 0)) )) = 0;
```

 

```
  wr4: SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
      'PROCESS_PLANNING_SCHEMA.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) )
       | ('PROCESS_PLANNING_SCHEMA.ORIENTED_CLOSED_SHELL' IN
       TYPEOF(msb\manifold_solid_brep.outer)) )) = 0;
  wr5: SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
       'PROCESS_PLANNING_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 | (
      'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
      NOT ('PROCESS_PLANNING_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION'
       IN TYPEOF(mi\mapped_item.mapping_source.
       mapped_representation))) )) = 0;
END_ENTITY; -- advanced_brep_shape_representation

ENTITY advanced_face
 SUBTYPE OF (face_surface);
 WHERE
  wr1 : SIZEOF(['PROCESS_PLANNING_SCHEMA.ELEMENTARY_SURFACE',
       'PROCESS_PLANNING_SCHEMA.B_SPLINE_SURFACE',
       'PROCESS_PLANNING_SCHEMA.SWEPT_SURFACE'] * TYPEOF(
       face_geometry)) = 1;
  wr2 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
       'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
        | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
       edge_list | (NOT ('PROCESS_PLANNING_SCHEMA.EDGE_CURVE' IN
       TYPEOF(oe\oriented_edge.edge_element))) )) = 0)) )) = 0;
  wr3 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
       'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
        | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
       edge_list | (NOT (SIZEOF(['PROCESS_PLANNING_SCHEMA.LINE',
       'PROCESS_PLANNING_SCHEMA.CONIC',
       'PROCESS_PLANNING_SCHEMA.POLYLINE',
       'PROCESS_PLANNING_SCHEMA.SURFACE_CURVE',
       'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(oe.
       edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
        0;
  wr4 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
       'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
        | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
       edge_list | (NOT ((('PROCESS_PLANNING_SCHEMA.VERTEX_POINT'
       IN TYPEOF(oe\edge.edge_start)) AND (
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(oe\edge
       .edge_start\vertex_point.vertex_geometry))) AND ((
       'PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(oe\edge.
       edge_end)) AND ('PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT'
      IN TYPEOF(oe\edge.edge_end\vertex_point.vertex_geometry))))) ))
       = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
       'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
        | ('PROCESS_PLANNING_SCHEMA.ORIENTED_PATH' IN TYPEOF(
       elp_fbnds.bound)) )) = 0;
  wr6 : (NOT ('PROCESS_PLANNING_SCHEMA.SWEPT_SURFACE' IN TYPEOF(
       face_geometry))) OR (SIZEOF(['PROCESS_PLANNING_SCHEMA.LINE',
       'PROCESS_PLANNING_SCHEMA.CONIC',
       'PROCESS_PLANNING_SCHEMA.POLYLINE',
       'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE'] * TYPEOF(
       face_geometry\swept_surface.swept_curve)) = 1);
  wr7 : SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* bounds | (
       'PROCESS_PLANNING_SCHEMA.VERTEX_LOOP' IN TYPEOF(bnds.bound)) )
        | (NOT (('PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(
```

```
        vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND (
        'PROCESS_PLANNING_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([
        'PROCESS_PLANNING_SCHEMA.EDGE_LOOP',
        'PROCESS_PLANNING_SCHEMA.VERTEX_LOOP'] * TYPEOF(bnd.bound))
        = 1)) )) = 0;
   wr9 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
        'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
         | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
        edge_list | (('PROCESS_PLANNING_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 (
       'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(sc_ag))) )) = 0))) ))
        = 0)) )) = 0;
   wr10: ((NOT ('PROCESS_PLANNING_SCHEMA.SWEPT_SURFACE' IN TYPEOF(
        face_geometry))) OR ((NOT (
        'PROCESS_PLANNING_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 | (
        'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
         | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path.
        edge_list | (('PROCESS_PLANNING_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

ENTITY allowed_time
 SUBTYPE OF (action_property);
 WHERE
 wr1: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(
       it)) = 2) AND (it.name = 'allowance factor')) )) = 1) )) = 0);
  wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(
       it)) = 2) AND (it.name = 'standard time')) )) = 1) )) = 0);
  wr3: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'allowed type')) )) <= 1) )) = 0;
             w r 4 :     S I Z E O F ( Q U E R Y     (     a p r     < *
USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'allowed time source')) )) = 1) ))
       = 1;
END_ENTITY; -- allowed_time
```

 

```
ENTITY alternate_action_method_relationship
 SUBTYPE OF (action_method_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
       related_method);
  wr2: 'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
       relating_method);
END_ENTITY; -- alternate_action_method_relationship

ENTITY alternate_plan_relationship
 SUBTYPE OF (action_relationship);
 WHERE
  wr1: ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(SELF.
       relating_action)) AND (
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(
       SELF.related_action));
  wr2: SIZEOF(QUERY ( edir <* USEDIN(SELF.related_action,
       'PROCESS_PLANNING_SCHEMA.ID_ATTRIBUTE.IDENTIFIED_ITEM') | (
       edir.attribute_value = 'alternate process plan') )) = 1;
END_ENTITY; -- alternate_plan_relationship

ENTITY ancillary_activity
 SUBTYPE OF (manufacturing_activity);
END_ENTITY; -- ancillary_activity

ENTITY ancillary_setup
 SUBTYPE OF (process_plan_activity);
END_ENTITY; -- ancillary_setup

ENTITY angular_location
 SUBTYPE OF (dimensional_location);
  angle_selection : angle_relator;
END_ENTITY; -- angular_location

ENTITY angular_size
 SUBTYPE OF (dimensional_size);
  angle_selection : angle_relator;
END_ENTITY; -- angular_size

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

ENTITY apex
 SUBTYPE OF (derived_shape_aspect);
END_ENTITY; -- apex

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,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
```

```
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- application_context

ENTITY application_context_element
 SUPERTYPE OF (ONEOF (product_context,product_definition_context));
  name        : label;
  frame_of_reference : application_context;
END_ENTITY; -- application_context_element

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

ENTITY applied_action_assignment
 SUBTYPE OF (action_assignment);
  items : SET [1:?] OF action_item;
END_ENTITY; -- applied_action_assignment

ENTITY applied_action_method_assignment
 SUBTYPE OF (action_method_assignment);
  items : SET [1:?] OF action_method_item;
END_ENTITY; -- applied_action_method_assignment

ENTITY applied_action_request_assignment
 SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF action_request_item;
END_ENTITY; -- applied_action_request_assignment

ENTITY applied_approval_assignment
 SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF approval_item;
END_ENTITY; -- applied_approval_assignment

ENTITY applied_area
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(
       SELF.of_shape);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
```

 

```
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
        + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'effective length')) )) = 1)) ))
        = 0)) )) <= 1;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'maximum length')) )) <= 1)) ))
        = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
        name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
END_ENTITY; -- applied_area

ENTITY applied_classification_assignment
 SUBTYPE OF (classification_assignment);
  items : SET [1:?] OF classification_item;
END_ENTITY; -- applied_classification_assignment

ENTITY applied_contract_assignment
 SUBTYPE OF (contract_assignment);
  items : SET [1:?] OF contract_item;
END_ENTITY; -- applied_contract_assignment

ENTITY applied_date_assignment
 SUBTYPE OF (date_assignment);
  items : SET [1:?] OF date_item;
```

```
END_ENTITY; -- applied_date_assignment

ENTITY applied_document_reference
 SUBTYPE OF (document_reference);
  items : SET [1:?] OF document_reference_item;
END_ENTITY; -- applied_document_reference

ENTITY applied_document_usage_constraint_assignment
 SUBTYPE OF (document_usage_constraint_assignment);
  items : SET [1:?] OF document_usage_constraint_item;
END_ENTITY; -- applied_document_usage_constraint_assignment

ENTITY applied_effectivity_assignment
 SUBTYPE OF (effectivity_assignment);
  items : SET [1:?] OF effectivity_item;
END_ENTITY; -- applied_effectivity_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

ENTITY applied_group_assignment
 SUBTYPE OF (group_assignment);
  items : SET [1:?] OF group_item;
END_ENTITY; -- applied_group_assignment

ENTITY applied_identification_assignment
 SUBTYPE OF (identification_assignment);
  items : SET [1:?] OF identification_assignment_item;
END_ENTITY; -- applied_identification_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 (
       'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_CLASS' IN
       TYPEOF(SELF\classification_assignment.assigned_class));
  wr3: SIZEOF(QUERY ( edir <* USEDIN(SELF.assigned_class,
'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATED_ITEM')
       | (((edir.name = 'name scope') AND
      ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_GENERAL_PROPERTY'
       IN TYPEOF(edir.relating_item))) AND (SIZEOF(QUERY ( gpa <*
       USEDIN(edir.relating_item,
  'PROCESS_PLANNING_SCHEMA.GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION')
       | ((gpa.name = 'definitional') AND (SIZEOF(QUERY ( pdr <*
       USEDIN(QUERY ( pd <* gpa.derived_definition.
       derived_property_select | (
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION' IN TYPEOF(pd))
 ),'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | ((pdr.used_representation.name = 'property value') AND (
       SIZEOF(QUERY ( ri <* pdr.used_representation.items | (
       'PROCESS_PLANNING_SCHEMA.VALUE_REPRESENTATION_ITEM' IN
       TYPEOF(ri)) )) >= 1)) )) = 1)) )) = 1)) )) = 1;
END_ENTITY; -- applied_library_assignment
```

 

```
ENTITY applied_organization_assignment
 SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF organization_item;
END_ENTITY; -- applied_organization_assignment

ENTITY applied_security_classification_assignment
 SUBTYPE OF (security_classification_assignment);
  items : SET [1:?] OF security_classification_item;
END_ENTITY; -- applied_security_classification_assignment

ENTITY approval;
  status : approval_status;
  level : label;
END_ENTITY; -- approval

ENTITY approval_assignment
 ABSTRACT SUPERTYPE;
  assigned_approval : approval;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_assignment

ENTITY approval_date_time;
  date_time   : date_time_select;
  dated_approval : approval;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- approval_date_time

ENTITY approval_person_organization;
  person_organization : person_organization_select;
  authorized_approval : approval;
  role        : approval_role;
END_ENTITY; -- approval_person_organization

ENTITY approval_role;
  role : label;
 DERIVE
  description : text := get_description_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- approval_role

ENTITY approval_status;
  name : label;
END_ENTITY; -- approval_status

ENTITY assembly_component_usage
 SUPERTYPE OF (next_assembly_usage_occurrence)
 SUBTYPE OF (product_definition_usage);
  reference_designator : OPTIONAL identifier;
END_ENTITY; -- assembly_component_usage

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

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

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

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: ((('PROCESS_PLANNING_SCHEMA.UNIFORM_CURVE' IN TYPEOF(SELF)) OR
       ('PROCESS_PLANNING_SCHEMA.QUASI_UNIFORM_CURVE' IN TYPEOF(
       SELF))) OR ('PROCESS_PLANNING_SCHEMA.BEZIER_CURVE' IN
       TYPEOF(SELF))) OR (
       'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE_WITH_KNOTS' IN
       TYPEOF(SELF));
END_ENTITY; -- b_spline_curve

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

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: ((('PROCESS_PLANNING_SCHEMA.UNIFORM_SURFACE' IN TYPEOF(SELF))
       OR ('PROCESS_PLANNING_SCHEMA.QUASI_UNIFORM_SURFACE' IN
       TYPEOF(SELF))) OR ('PROCESS_PLANNING_SCHEMA.BEZIER_SURFACE'
       IN TYPEOF(SELF))) OR (
       'PROCESS_PLANNING_SCHEMA.B_SPLINE_SURFACE_WITH_KNOTS' IN
       TYPEOF(SELF));
END_ENTITY; -- b_spline_surface

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

ENTITY bezier_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY; -- bezier_curve
```

```
ENTITY bezier_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY; -- bezier_surface

ENTITY block_shape_representation
 SUBTYPE OF (shape_representation_with_parameters);
 WHERE
  wr1: SIZEOF(SELF.items) = 4;
  wr2: SIZEOF(QUERY ( it <* SELF.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1;
  wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
  wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1;
  wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'height')) )) = 1;
END_ENTITY; -- block_shape_representation

ENTITY boss
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SELF\characterized_object.description IN ['circular','complex',
       'rectangular'];
  wr2 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
       .description = 'boss height occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'path feature component usage') AND ((
       'PROCESS_PLANNING_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP')
       IN TYPEOF(sar))) ) | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
```

 

```
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
        .description = 'circular profile occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (
        'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
        TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
  wr7 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
        .description = 'top condition occurrence') AND (SIZEOF(
        QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
         | ((sar.description = 'boss top usage') AND (
        'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
         TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.BOSS_TOP' IN
        TYPEOF(fcr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
  wr8 : (NOT (SELF\characterized_object.description = 'circular')) OR
        (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
        .description = 'change in diameter occurrence') AND (
        SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
        AND (
        'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
         TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.TAPER' IN
        TYPEOF(fcr.related_shape_aspect)) AND (
        'PROCESS_PLANNING_SCHEMA.BOSS' IN TYPEOF(fcr.
```

```
      relating_shape_aspect))) )) = 1)) )) <= 1)) )) = 0);
  wr9 : (NOT (SELF\characterized_object.description = 'complex')) OR (
      SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'enclosed boundary occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr
      .relating_shape_aspect)) = 1) )) = 1)) )) = 1)) )) = 0);
  wr10: (NOT (SELF\characterized_object.description IN ['complex',
      'rectangular'])) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'change in boundary occurrence') AND (
      SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | ((sar.description =
      'taper usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ((('PROCESS_PLANNING_SCHEMA.TAPER' IN
      TYPEOF(fcr.related_shape_aspect)) AND (
      'PROCESS_PLANNING_SCHEMA.BOSS' IN TYPEOF(fcr.
      relating_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'rectangular profile occurrence') AND (
      SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
  wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) | ((
      'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'maximum feature limit')) )) >=
      0;
END_ENTITY; -- boss

ENTITY boss_top
 SUBTYPE OF (shape_aspect);
```

 

```
WHERE
 wr1: 'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
 wr4: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'PROCESS_PLANNING_SCHEMA.LOCATION_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
 wr5: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
 wr6: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | (((sar.description =
      'boss top usage') AND (sar.name IN ['boss height start',
      'boss height end'])) AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
      'top condition occurrence') AND (
      'PROCESS_PLANNING_SCHEMA.BOSS' IN TYPEOF(fcr.
      related_shape_aspect.of_shape.definition))) AND (
      'PROCESS_PLANNING_SCHEMA.BOSS_TOP' IN TYPEOF(fcr.
      relating_shape_aspect))) )) >= 1;
 wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) <= 1)) )) = 0;
 wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
```

```
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'top radius')) )) <= 1)) ))
       = 0)) )) = 0;
END_ENTITY; -- boss_top

ENTITY boundary_curve
 SUBTYPE OF (composite_curve_on_surface);
 WHERE
  wr1: SELF\composite_curve.closed_curve;
END_ENTITY; -- boundary_curve

ENTITY bounded_curve
 SUPERTYPE OF (ONEOF (polyline,b_spline_curve,trimmed_curve,
   bounded_pcurve,bounded_surface_curve,composite_curve))
 SUBTYPE OF (curve);
END_ENTITY; -- bounded_curve

ENTITY bounded_pcurve
 SUBTYPE OF (pcurve, bounded_curve);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE' IN TYPEOF(SELF\pcurve.
       reference_to_curve.items[1]);
END_ENTITY; -- bounded_pcurve

ENTITY bounded_surface
 SUPERTYPE OF (ONEOF (b_spline_surface,rectangular_trimmed_surface,
   curve_bounded_surface,rectangular_composite_surface))
 SUBTYPE OF (surface);
END_ENTITY; -- bounded_surface

ENTITY bounded_surface_curve
 SUBTYPE OF (surface_curve, bounded_curve);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE' IN TYPEOF(SELF\
       surface_curve.curve_3d);
END_ENTITY; -- bounded_surface_curve

ENTITY brep_with_voids
 SUBTYPE OF (manifold_solid_brep);
  voids : SET [1:?] OF oriented_closed_shell;
END_ENTITY; -- brep_with_voids

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

ENTITY cartesian_point
 SUBTYPE OF (point);
  coordinates : LIST [1:3] OF length_measure;
END_ENTITY; -- cartesian_point
```

 

```
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

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

ENTITY centre_of_symmetry
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(QUERY ( sadr <* deriving_relationships | (NOT (
       'PROCESS_PLANNING_SCHEMA.SYMMETRIC_SHAPE_ASPECT' IN TYPEOF(
       sadr.related_shape_aspect))) )) = 0;
END_ENTITY; -- centre_of_symmetry

ENTITY chamfer
 SUBTYPE OF (transition_feature);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) ) | ((('PROCESS_PLANNING_SCHEMA.CHAMFER_OFFSET'
       IN TYPEOF(fcr.related_shape_aspect)) AND (
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) ) | ((('PROCESS_PLANNING_SCHEMA.CHAMFER_OFFSET'
       IN TYPEOF(fcr.related_shape_aspect)) AND (
       'PROCESS_PLANNING_SCHEMA.CHAMFER' IN TYPEOF(fcr.
       relating_shape_aspect))) AND (fcr.related_shape_aspect.
```

```
       description = 'second offset')) )) = 1;
END_ENTITY; -- chamfer

ENTITY chamfer_offset
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: SELF.description IN ['first offset','second offset'];
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'offset amount')) OR ((
```

 

```
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | (('PROCESS_PLANNING_SCHEMA.' +
      'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) ) | ((
      'PROCESS_PLANNING_SCHEMA.CHAMFER' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (
      'PROCESS_PLANNING_SCHEMA.CHAMFER_OFFSET' IN TYPEOF(sdr.
      related_shape_aspect))) )) = 1;
END_ENTITY; -- chamfer_offset

ENTITY characterized_object;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- characterized_object

ENTITY circle
 SUBTYPE OF (conic);
  radius : positive_length_measure;
END_ENTITY; -- circle

ENTITY circular_closed_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
      TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
```

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
        name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1)) )) = 0)) ))
        = 0;
END_ENTITY; -- circular_closed_profile

ENTITY circular_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') | (
        'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN
        TYPEOF(sdr.related_shape_aspect)) )) = 1) )) <= 3)) )) = 0;
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

 

```
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) <= 1)) )) =
      0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'base feature rotation')) ))
      <= 1)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND ('PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'angular spacing')) )) = 1)) ))
      = 0)) )) = 0;
  wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
      name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
END_ENTITY; -- circular_pattern

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

ENTITY class
 SUBTYPE OF (group);
END_ENTITY; -- class

ENTITY classification_assignment
 ABSTRACT SUPERTYPE;
  assigned_class : group;
  role      : classification_role;
END_ENTITY; -- classification_assignment

ENTITY classification_role;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- classification_role

ENTITY closed_path_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
       'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
       'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'PROCESS_PLANNING_SCHEMA.PATH_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) )) = 1)) )) = 0;
END_ENTITY; -- closed_path_profile
```

 

```
ENTITY closed_shell
 SUBTYPE OF (connected_face_set);
END_ENTITY; -- closed_shell

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

ENTITY common_datum
 SUBTYPE OF (composite_shape_aspect, datum);
 WHERE
  wr1: SIZEOF(SELF.component_relationships) = 2;
  wr2: SIZEOF(QUERY ( sar <* SELF.component_relationships | (NOT ((
       'PROCESS_PLANNING_SCHEMA.DATUM' IN TYPEOF(sar.
       related_shape_aspect)) AND (NOT (
       'PROCESS_PLANNING_SCHEMA.COMMON_DATUM' IN TYPEOF(sar.
       related_shape_aspect))))) )) = 0;
END_ENTITY; -- common_datum

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

ENTITY composite_curve_on_surface
 SUPERTYPE OF (boundary_curve)
 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

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: 'PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE' IN TYPEOF(parent_curve);
END_ENTITY; -- composite_curve_segment

ENTITY composite_hole
 SUBTYPE OF (compound_feature);
 WHERE
```

```
  wr1: SELF\characterized_object.description IN ['counterbore',
       'countersunk'];
  wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
       'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
       csa)) AND (SIZEOF(QUERY ( sar <* csa.component_relationships
       | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) AND ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
       'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
       csa)) AND (SIZEOF(QUERY ( sar <* csa.component_relationships
       | (('PROCESS_PLANNING_SCHEMA.ROUND_HOLE' IN TYPEOF(sar.
       related_shape_aspect)) AND (NOT (SIZEOF(QUERY ( pds <*
       QUERY ( pd <* USEDIN(sar.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'change in diameter occurrence') AND (SIZEOF(
       QUERY ( fcr2 <* QUERY ( sar2 <* USEDIN(sa_occ,
'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
       | ((sar2.description = 'taper usage') AND (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar2))) ) | ('PROCESS_PLANNING_SCHEMA.TAPER' IN
     TYPEOF(fcr2.related_shape_aspect)) )) = 1)) )) = 0)) )) = 0))) ))
       = 1)) )) = 1)) )) = 1);
END_ENTITY; -- composite_hole

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

ENTITY compound_feature
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pds <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((csa.
       name = 'compound feature in solid') AND (
       'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
       csa))) )) = 1)) )) = 1;
  wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
```

 

```
        'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
        csa)) )) = 1)) )) = 1;
    wr3: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
        csa)) AND (SIZEOF(QUERY ( fcr <* csa.component_relationships
        | (NOT (
        'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(fcr))) )) = 0)) )) = 1)) )) = 1;
    wr4: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (
        'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
        csa)) )) = 1)) )) = 1;
    wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
        csa)) AND (SIZEOF(QUERY ( sar <* csa.component_relationships
        | ('PROCESS_PLANNING_SCHEMA.THREAD' IN TYPEOF(sar.
        related_shape_aspect)) )) = 0)) )) = 1)) )) = 1;
    wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((
        'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
        csa)) AND (SIZEOF(QUERY ( sar <* csa.component_relationships
        | (('PROCESS_PLANNING_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

ENTITY compound_representation_item
 SUBTYPE OF (representation_item);
  item_element : compound_item_definition;
END_ENTITY; -- compound_representation_item

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

ENTITY conic
 SUPERTYPE OF (ONEOF (circle,ellipse,hyperbola,parabola))
 SUBTYPE OF (curve);
  position : axis2_placement;
END_ENTITY; -- conic

ENTITY conical_surface
 SUBTYPE OF (elementary_surface);
```

```
  radius   : length_measure;
  semi_angle : plane_angle_measure;
 WHERE
  wr1: radius >= 0;
END_ENTITY; -- conical_surface

ENTITY connected_edge_set
 SUBTYPE OF (topological_representation_item);
  ces_edges : SET [1:?] OF edge;
END_ENTITY; -- connected_edge_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

ENTITY context_dependent_unit
 SUBTYPE OF (named_unit);
  name : label;
END_ENTITY; -- context_dependent_unit

ENTITY continuous_process_relationship
 SUBTYPE OF (sequential_method);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       related_method);
  wr2: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       relating_method);
 wr3: (SELF.description IN ['serial','batch', 'serial and batch']);
END_ENTITY; -- continuous_process_relationship

ENTITY contract;
  name  : label;
  purpose : text;
  kind  : contract_type;
END_ENTITY; -- contract

ENTITY contract_assignment
 ABSTRACT SUPERTYPE;
  assigned_contract : contract;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- contract_assignment

ENTITY contract_type;
  description : label;
END_ENTITY; -- contract_type

ENTITY controller
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
       TYPEOF(adr.assigned_document)) )) = 1;
END_ENTITY; -- controller
```

 

```
ENTITY controller_program
 SUBTYPE OF (document);
 WHERE
  wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')
       | (duc.subject_element =
       'controller program revision level') )) = 1;
  wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (SIZEOF(QUERY ( d <* adr.items | (((
       'PROCESS_PLANNING_SCHEMA.DATE_AND_TIME' IN TYPEOF(d)) AND (
       'PROCESS_PLANNING_SCHEMA.DATE' IN TYPEOF(d.date_component)))
       AND ('PROCESS_PLANNING_SCHEMA.LOCAL_TIME' IN TYPEOF(d.
       time_component))) )) = 1) )) = 1;
  wr3: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (SIZEOF(QUERY ( d <* adr.items | (
     'PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY' IN TYPEOF(d)) ))
       >= 1) )) = 1;
  wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(adr)) AND (NOT (SIZEOF(QUERY ( pdf <* adr.items | ((
       ('PROCESS_PLANNING_SCHEMA.' + 'PRODUCT_DEFINITION_FORMATION')
       IN TYPEOF(pdf)) AND (NOT (SIZEOF(QUERY ( pd <* USEDIN(pdf,
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION.FORMATION') | (
       pd.frame_of_reference.name = 'part definition') )) = 1))) ))
       = 1))) )) = 0;
  wr5: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (SIZEOF(QUERY ( d <* adr.items | (
       'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(d)) )) >=
       1) )) = 1;
  wr6: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (SIZEOF(QUERY ( d <* adr.items | (
      'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(d)) ))
       >= 1) )) = 1;
END_ENTITY; -- controller_program

ENTITY conversion_based_unit
 SUBTYPE OF (named_unit);
  name       : label;
  conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit

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

ENTITY curve
 SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve,offset_curve_3d,
   curve_replica))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- curve

ENTITY curve_bounded_surface
 SUBTYPE OF (bounded_surface);
  basis_surface : surface;
  boundaries   : SET [1:?] OF boundary_curve;
  implicit_outer : BOOLEAN;
 WHERE
  wr1: (NOT implicit_outer) OR (SIZEOF(QUERY ( temp <* boundaries | (
       'PROCESS_PLANNING_SCHEMA.OUTER_BOUNDARY_CURVE' IN TYPEOF(
       temp)) )) = 0);
  wr2: (NOT implicit_outer) OR (
       'PROCESS_PLANNING_SCHEMA.BOUNDED_SURFACE' IN TYPEOF(
       basis_surface));
  wr3: SIZEOF(QUERY ( temp <* boundaries | (
       'PROCESS_PLANNING_SCHEMA.OUTER_BOUNDARY_CURVE' IN TYPEOF(
       temp)) )) <= 1;
  wr4: SIZEOF(QUERY ( temp <* boundaries | (temp\
       composite_curve_on_surface.basis_surface[1] <> SELF.
       basis_surface) )) = 0;
END_ENTITY; -- curve_bounded_surface

ENTITY curve_replica
 SUBTYPE OF (curve);
  parent_curve  : curve;
  transformation : cartesian_transformation_operator;
 WHERE
  wr1: transformation.dim = parent_curve.dim;
  wr2: acyclic_curve_replica(SELF,parent_curve);
END_ENTITY; -- curve_replica

ENTITY cylindrical_shape_representation
 SUBTYPE OF (shape_representation_with_parameters);
 WHERE
  wr1: SIZEOF(SELF.items) = 3;
  wr2: SIZEOF(QUERY ( it <* SELF.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1;
  wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
  wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) = 1;
END_ENTITY; -- cylindrical_shape_representation

ENTITY cylindrical_surface
 SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- cylindrical_surface

ENTITY cylindricity_tolerance
 SUBTYPE OF (geometric_tolerance);
```

 

```
 WHERE
  wr1: NOT (('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- cylindricity_tolerance

ENTITY data_environment;
  name    : label;
  description : text;
  elements  : SET [1:?] OF property_definition_representation;
END_ENTITY; -- data_environment

ENTITY date
 SUPERTYPE OF (calendar_date);
  year_component : year_number;
END_ENTITY; -- date

ENTITY date_and_time;
  date_component : date;
  time_component : local_time;
END_ENTITY; -- date_and_time

ENTITY date_assignment
 ABSTRACT SUPERTYPE;
  assigned_date : date;
  role     : date_role;
END_ENTITY; -- date_assignment

ENTITY date_role;
  name : label;
 DERIVE
  description : text := get_description_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- date_role

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.relating_shape_aspect) * [
       'PROCESS_PLANNING_SCHEMA.DATUM_FEATURE',
       'PROCESS_PLANNING_SCHEMA.DATUM_TARGET']) <> 1) )) = 0;
END_ENTITY; -- datum

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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT')) | (NOT (
       'PROCESS_PLANNING_SCHEMA.DATUM' IN TYPEOF(sar.
       related_shape_aspect))) )) = 0;
```

```
  wr2: SELF.product_definitional = TRUE;
END_ENTITY; -- datum_feature

ENTITY datum_reference;
  precedence    : INTEGER;
  referenced_datum : datum;
 WHERE
  wr1: precedence > 0;
END_ENTITY; -- datum_reference

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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT')) | (NOT (
       'PROCESS_PLANNING_SCHEMA.DATUM' IN TYPEOF(sar.
       related_shape_aspect))) )) = 0;
  wr2: SELF.product_definitional = TRUE;
END_ENTITY; -- datum_target

ENTITY defining_action_method_relationship
 SUBTYPE OF (action_method_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(SELF.
       related_method);
END_ENTITY; -- defining_action_method_relationship

ENTITY definitional_representation
 SUBTYPE OF (representation);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PARAMETRIC_REPRESENTATION_CONTEXT' IN
       TYPEOF(SELF\representation.context_of_items);
END_ENTITY; -- definitional_representation

ENTITY degenerate_pcurve
 SUBTYPE OF (point);
  basis_surface   : surface;
  reference_to_curve : definitional_representation;
 WHERE
  wr1: SIZEOF(reference_to_curve\representation.items) = 1;
  wr2: 'PROCESS_PLANNING_SCHEMA.CURVE' IN TYPEOF(reference_to_curve\
       representation.items[1]);
  wr3: reference_to_curve\representation.items[1]\
       geometric_representation_item.dim = 2;
END_ENTITY; -- degenerate_pcurve

ENTITY degenerate_toroidal_surface
 SUBTYPE OF (toroidal_surface);
  select_outer : BOOLEAN;
 WHERE
  wr1: major_radius < minor_radius;
END_ENTITY; -- degenerate_toroidal_surface

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

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,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- derived_unit

ENTITY derived_unit_element;
  unit   : named_unit;
  exponent : REAL;
END_ENTITY; -- derived_unit_element

ENTITY description_attribute;
  attribute_value : text;
  described_item : description_attribute_select;
END_ENTITY; -- description_attribute

ENTITY descriptive_representation_item
 SUBTYPE OF (representation_item);
  description : text;
END_ENTITY; -- descriptive_representation_item

ENTITY design_reference
 SUBTYPE OF (document);
 WHERE
  wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')
       | (duc.subject_element = 'drawing revision level') )) = 1;
  wr2: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (NOT (SIZEOF(QUERY ( d <* adr.items | (
      'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) ))
       >= 1)) )) = 0;
END_ENTITY; -- design_reference

ENTITY dimension_related_tolerance_zone_element;
  related_dimension : dimensional_location;
  related_element  : tolerance_zone_definition;
END_ENTITY; -- dimension_related_tolerance_zone_element

ENTITY dimensional_characteristic_representation;
  dimension   : dimensional_characteristic;
  representation : shape_dimension_representation;
END_ENTITY; -- dimensional_characteristic_representation

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

ENTITY dimensional_location
 SUPERTYPE OF (ONEOF (angular_location,dimensional_location_with_path))
 SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- dimensional_location

ENTITY dimensional_location_with_path
 SUBTYPE OF (dimensional_location);
  path : shape_aspect;
END_ENTITY; -- dimensional_location_with_path

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

ENTITY dimensional_size_with_path
 SUBTYPE OF (dimensional_size);
  path : shape_aspect;
END_ENTITY; -- dimensional_size_with_path

ENTITY directed_action
 SUBTYPE OF (executed_action);
  directive : action_directive;
END_ENTITY; -- directed_action

ENTITY directed_dimensional_location
 SUBTYPE OF (dimensional_location);
END_ENTITY; -- directed_dimensional_location

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

ENTITY direction_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(SELF.items) = 1;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
       'PROCESS_PLANNING_SCHEMA.DIRECTION' IN TYPEOF(it))) )) = 0;
END_ENTITY; -- direction_shape_representation

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

ENTITY document_file
 SUBTYPE OF (document, characterized_object);
 WHERE
  wr1: SIZEOF(QUERY ( drt <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT')
       | (drt.name IN ['physical','digital']) )) = 1;
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_IDENTIFICATION_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.IDENTIFICATION_ASSIGNMENT' IN
       TYPEOF(adr)) AND (adr.role.name = 'version')) )) <= 1;
  wr3: SIZEOF(QUERY ( aeia <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.EXTERNAL_IDENTIFICATION_ASSIGNMENT'
       IN TYPEOF(aeia)) )) >= 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ((pd.name = 'document property') AND (SIZEOF(QUERY ( pdr <*
       USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | (('PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE_PROPERTIES' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'document format')) )) <= 1)) ))
       <= 1;
END_ENTITY; -- document_file

ENTITY document_file_properties
 SUBTYPE OF (representation);
 WHERE
  wr1 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'country code')
       AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr2 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'detail level')
       AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr3 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'geometry type')
       AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr4 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'language code')
       AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr5 : SIZEOF(QUERY ( r <* SELF.items | ((r.name =
       'creating interface') AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr6 : SIZEOF(QUERY ( r <* SELF.items | ((r.name =
       'creating operating system') AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr7 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'creating system')
       AND (
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(r))) )) <= 1;
  wr8 : SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'data format') AND
        ('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
```

```
      IN TYPEOF(r))) )) <= 1;
  wr9 : SIZEOF(QUERY ( r <* SELF.items | ((r.name =
      'format character code') AND (
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(r))) )) <= 1;
  wr10: SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'file size') AND (
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
      TYPEOF(r)) = 2)) )) <= 1;
  wr11: SIZEOF(QUERY ( r <* SELF.items | ((r.name = 'page count') AND
      (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.COUNT_MEASURE'] * TYPEOF(r)) = 2)) ))
      <= 1;
END_ENTITY; -- document_file_properties

ENTITY document_product_association;
  name       : label;
  description    : OPTIONAL text;
  relating_document : document;
  related_product  : product_or_formation_or_definition;
END_ENTITY; -- document_product_association

ENTITY document_reference
 ABSTRACT SUPERTYPE;
  assigned_document : document;
  source      : label;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
      'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- document_reference

ENTITY document_relationship;
  name       : label;
  description    : OPTIONAL text;
  relating_document : document;
  related_document : document;
END_ENTITY; -- document_relationship

ENTITY document_representation_type;
  name         : label;
  represented_document : document;
END_ENTITY; -- document_representation_type

ENTITY document_type;
  product_data_type : label;
END_ENTITY; -- document_type

ENTITY document_usage_constraint;
  source        : document;
  subject_element    : label;
  subject_element_value : text;
END_ENTITY; -- document_usage_constraint

ENTITY document_usage_constraint_assignment
 ABSTRACT SUPERTYPE;
  assigned_document_usage : document_usage_constraint;
  role          : document_usage_role;
```

 

```
END_ENTITY; -- document_usage_constraint_assignment

ENTITY document_usage_role;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- document_usage_role

ENTITY edge
 SUPERTYPE OF (ONEOF (edge_curve,oriented_edge))
 SUBTYPE OF (topological_representation_item);
  edge_start : vertex;
  edge_end  : vertex;
END_ENTITY; -- edge

ENTITY edge_based_wireframe_model
 SUBTYPE OF (geometric_representation_item);
  ebwm_boundary : SET [1:?] OF connected_edge_set;
END_ENTITY; -- edge_based_wireframe_model

ENTITY edge_based_wireframe_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       >= 1;
  wr3: SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
       edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
       QUERY ( edges <* eb.ces_edges | (NOT (
       'PROCESS_PLANNING_SCHEMA.EDGE_CURVE' IN TYPEOF(edges))) )) =
       0)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
       edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
       QUERY ( pline_edges <* QUERY ( edges <* eb.ces_edges | (
       'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(edges\
       edge_curve.edge_geometry)) ) | (NOT (SIZEOF(pline_edges\
       edge_curve.edge_geometry\polyline.points) > 2)) )) = 0)) ))
       = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
       edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
       QUERY ( edges <* eb.ces_edges | (NOT ((
       'PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(edges.
       edge_start)) AND ('PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN
       TYPEOF(edges.edge_end)))) )) = 0)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
       edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
       QUERY ( edges <* eb.ces_edges | (NOT
       valid_wireframe_edge_curve(edges\edge_curve.edge_geometry)) ))
```

```
      = 0)) )) = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( ebwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( eb <* ebwm\
       edge_based_wireframe_model.ebwm_boundary | (NOT (SIZEOF(
       QUERY ( edges <* eb.ces_edges | (NOT (
       valid_wireframe_vertex_point(edges.edge_start\vertex_point.
       vertex_geometry) AND valid_wireframe_vertex_point(edges.
       edge_end\vertex_point.vertex_geometry))) )) = 0)) )) = 0)) ))
       = 0;
  wr8: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
       NOT (('PROCESS_PLANNING_SCHEMA.' +
       'EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
       mapped_item.mapping_source.mapped_representation))) )) = 0;
  wr9: SELF.context_of_items\geometric_representation_context.
       coordinate_space_dimension = 3;
END_ENTITY; -- edge_based_wireframe_shape_representation

ENTITY edge_curve
 SUBTYPE OF (edge, geometric_representation_item);
  edge_geometry : curve;
  same_sense  : BOOLEAN;
END_ENTITY; -- edge_curve

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

ENTITY edge_round
 SUBTYPE OF (transition_feature);
 WHERE
  wr1: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
       SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

 

```
      ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
        ['PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
        = 0)) )) = 0);
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'second face shape')) )) <= 1)) ))
        = 0;
END_ENTITY; -- edge_round
```

```
ENTITY effectivity;
  id : identifier;
 DERIVE
  name    : label := get_name_value(SELF);
  description : text := get_description_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- effectivity

ENTITY effectivity_assignment
 ABSTRACT SUPERTYPE;
  assigned_effectivity : effectivity;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- effectivity_assignment

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

ENTITY ellipse
 SUBTYPE OF (conic);
  semi_axis_1 : positive_length_measure;
  semi_axis_2 : positive_length_measure;
END_ENTITY; -- ellipse

ENTITY evaluated_degenerate_pcurve
 SUBTYPE OF (degenerate_pcurve);
  equivalent_point : cartesian_point;
END_ENTITY; -- evaluated_degenerate_pcurve

ENTITY executed_action
 SUBTYPE OF (action);
END_ENTITY; -- executed_action

ENTITY extension
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- extension

ENTITY external_identification_assignment
 ABSTRACT SUPERTYPE
 SUBTYPE OF (identification_assignment);
  source : external_source;
END_ENTITY; -- external_identification_assignment

ENTITY external_source;
  source_id : source_item;
 DERIVE
  description : text := get_description_value(SELF);
 WHERE
```

 

```
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- external_source

ENTITY externally_defined_class
 SUBTYPE OF (externally_defined_item, class);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);
  wr2: SELF.source.name = 'ISO 13584 library';
  wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF.source,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
       | (aoa.role.name = 'library supplier') )) = 1;
  wr4: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')
       | ((
       'PROCESS_PLANNING_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
       IN TYPEOF(aoa)) AND (aoa.role.name = 'class version')) )) =
       1;
END_ENTITY; -- externally_defined_class

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,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | (adr.assigned_document.description =
       'externally defined dimension specification') )) <= 1;
END_ENTITY; -- externally_defined_dimension_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
       .description = 'partial area occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'applied area usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
```

```
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
      1)) )) = 0)) )) = 0);
  wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
      = 0)) )) = 0;
  wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'applied shape') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ('PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'reference presure angle')) )
     = 1)) )) = 0)) )) = 0);
wr20: (NOT (SELF\characterized_object.description IN ['gear'])) OR (
     SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'root fillet radius')) ))
        <= 1)) )) = 0)) )) = 0);
 END_ENTITY; -- externally_defined_feature_definition

 ENTITY externally_defined_general_property
  SUBTYPE OF (general_property, externally_defined_item);
  WHERE
    wr1: 'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);
    wr2: SELF.source.name = 'ISO 13584 library';
    wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS')
        | ((
        'PROCESS_PLANNING_SCHEMA.LIBRARY_CLASS_VERSION_ASSIGNMENT'
        IN TYPEOF(aoa)) AND (aoa.role.name = 'property version')) ))
        = 1;
    wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM')
        | ((ap.name = 'name scope') AND (
        'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_CLASS' IN
        TYPEOF(ap.related_item))) )) >= 1;
 END_ENTITY; -- externally_defined_general_property

 ENTITY externally_defined_item;
    item_id : source_item;
    source : external_source;
 END_ENTITY; -- externally_defined_item

 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
```

```
ENTITY externally_defined_representation_with_parameters
 SUBTYPE OF (representation);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_LIBRARY_ASSIGNMENT' IN
       TYPEOF(adr)) AND (adr.role.name =
       'definitional class membership')) )) = 1;
  wr2: SIZEOF(QUERY ( adr <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(adr)) )) <= 1;
  wr3: SIZEOF(QUERY ( adr <* SELF.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(adr)) AND (
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(adr.
       location))) )) <= 1;
END_ENTITY; -- externally_defined_representation_with_parameters

ENTITY externally_defined_schema
 SUBTYPE OF (externally_defined_item);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(SELF.source);
  wr2: SELF.source.name = 'ISO 10303 part';
  wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | (adr.assigned_document.description =
       'externally defined schema') )) <= 1;
  wr4: SELF.item_id IN ['externally defined schema','executable'];
END_ENTITY; -- externally_defined_schema

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 | (
       'PROCESS_PLANNING_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(temp)) ))
       <= 1;
END_ENTITY; -- face

ENTITY face_bound
 SUBTYPE OF (topological_representation_item);
  bound    : loop;
  orientation : BOOLEAN;
END_ENTITY; -- face_bound

ENTITY face_outer_bound
 SUBTYPE OF (face_bound);
END_ENTITY; -- face_outer_bound

ENTITY face_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(SELF.items) >= 1;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT ((
       'PROCESS_PLANNING_SCHEMA.FACE_SURFACE' IN TYPEOF(it)) OR (
       'PROCESS_PLANNING_SCHEMA.ORIENTED_FACE' IN TYPEOF(it)))) ))
       = 0;
END_ENTITY; -- face_shape_representation

ENTITY face_surface
 SUBTYPE OF (face, geometric_representation_item);
```

 

```
  face_geometry : surface;
  same_sense  : BOOLEAN;
 WHERE
  wr1: NOT ('PROCESS_PLANNING_SCHEMA.ORIENTED_SURFACE' IN TYPEOF(
       face_geometry));
END_ENTITY; -- face_surface

ENTITY faceted_brep
 SUBTYPE OF (manifold_solid_brep);
END_ENTITY; -- faceted_brep

ENTITY faceted_brep_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( it <* items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( it <* items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       > 0;
  wr3: SIZEOF(QUERY ( fbrep <* QUERY ( it <* items | (
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP' IN TYPEOF(it)) ) | (
       NOT (SIZEOF(QUERY ( csh <* msb_shells(fbrep) | (NOT (SIZEOF(
       QUERY ( fcs <* csh\connected_face_set.cfs_faces | (NOT ((
       'PROCESS_PLANNING_SCHEMA.FACE_SURFACE' IN TYPEOF(fcs)) AND (
       ('PROCESS_PLANNING_SCHEMA.PLANE' IN TYPEOF(fcs\face_surface.
       face_geometry)) AND (
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(fcs\
       face_surface.face_geometry\elementary_surface.position.
       location))))) )) = 0)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( fbrep <* QUERY ( it <* items | (
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP' IN TYPEOF(it)) ) | (
       NOT (SIZEOF(QUERY ( csh <* msb_shells(fbrep) | (NOT (SIZEOF(
       QUERY ( fcs <* csh\connected_face_set.cfs_faces | (NOT (
       SIZEOF(QUERY ( bnds <* fcs.bounds | (
       'PROCESS_PLANNING_SCHEMA.FACE_OUTER_BOUND' IN TYPEOF(bnds)) ))
       = 1)) )) = 0)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
      'PROCESS_PLANNING_SCHEMA.MANIFOLD_SOLID_BREP' IN TYPEOF(it)) )
       | ('PROCESS_PLANNING_SCHEMA.ORIENTED_CLOSED_SHELL' IN
       TYPEOF(msb\manifold_solid_brep.outer)) )) = 0;
  wr6: SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
       'PROCESS_PLANNING_SCHEMA.BREP_WITH_VOIDS' IN TYPEOF(it)) )
       | (NOT (SIZEOF(QUERY ( csh <* brv\brep_with_voids.voids |
       csh\oriented_closed_shell.orientation )) = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( mi <* QUERY ( it <* items | (
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
       NOT (
       'PROCESS_PLANNING_SCHEMA.FACETED_BREP_SHAPE_REPRESENTATION'
       IN TYPEOF(mi\mapped_item.mapping_source.
       mapped_representation))) )) = 0;
END_ENTITY; -- faceted_brep_shape_representation

ENTITY feature_component_definition
 SUBTYPE OF (characterized_object);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
```

```
       (NOT (SIZEOF(USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE')) = 1)) )) =
       0;
END_ENTITY; -- feature_component_definition

ENTITY feature_component_relationship
 SUPERTYPE OF (ONEOF (pattern_omit_membership,pattern_offset_membership))
 SUBTYPE OF (shape_aspect_relationship);
 WHERE
  wr1: ((SIZEOF(['PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT',
       'PROCESS_PLANNING_SCHEMA.REPLICATE_FEATURE',
       'PROCESS_PLANNING_SCHEMA.TRANSITION_FEATURE',
       'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN'] * TYPEOF(SELF.
       relating_shape_aspect)) = 1) OR (
       'PROCESS_PLANNING_SCHEMA.FEATURE_DEFINITION' IN TYPEOF(SELF.
       relating_shape_aspect.of_shape.definition))) OR (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.relating_shape_aspect.of_shape.definition));
END_ENTITY; -- feature_component_relationship

ENTITY feature_definition
 SUBTYPE OF (characterized_object);
 WHERE
  wr1: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
    SELF) | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF(pdr.used_representation)) )) <= 1;
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr3: SIZEOF(['PROCESS_PLANNING_SCHEMA.BOSS',
       'PROCESS_PLANNING_SCHEMA.TURNED_KNURL',
       'PROCESS_PLANNING_SCHEMA.THREAD',
       'PROCESS_PLANNING_SCHEMA.GEAR',
       'PROCESS_PLANNING_SCHEMA.MARKING',
       'PROCESS_PLANNING_SCHEMA.RIB_TOP',
       'PROCESS_PLANNING_SCHEMA.ROUND_HOLE',
       'PROCESS_PLANNING_SCHEMA.OUTSIDE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.POCKET',
       'PROCESS_PLANNING_SCHEMA.REMOVAL_VOLUME',
       'PROCESS_PLANNING_SCHEMA.REVOLVED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.OUTER_ROUND',
       'PROCESS_PLANNING_SCHEMA.FLAT_FACE',
       'PROCESS_PLANNING_SCHEMA.PROTRUSION',
       'PROCESS_PLANNING_SCHEMA.ROUNDED_END',
       'PROCESS_PLANNING_SCHEMA.SLOT',
       'PROCESS_PLANNING_SCHEMA.SPHERICAL_CAP',
       'PROCESS_PLANNING_SCHEMA.STEP',
       'PROCESS_PLANNING_SCHEMA.COMPOUND_FEATURE',
       'PROCESS_PLANNING_SCHEMA.REPLICATE_FEATURE',
     'PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
       * TYPEOF(SELF)) = 1;
  wr4: (NOT (SIZEOF(['PROCESS_PLANNING_SCHEMA.ROUND_HOLE',
       'PROCESS_PLANNING_SCHEMA.BOSS',
```

 

```
       'PROCESS_PLANNING_SCHEMA.OUTSIDE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.REMOVAL_VOLUME',
       'PROCESS_PLANNING_SCHEMA.FLAT_FACE',
       'PROCESS_PLANNING_SCHEMA.POCKET',
       'PROCESS_PLANNING_SCHEMA.PROTRUSION',
       'PROCESS_PLANNING_SCHEMA.RIB_TOP',
       'PROCESS_PLANNING_SCHEMA.ROUNDED_END',
       'PROCESS_PLANNING_SCHEMA.SLOT',
       'PROCESS_PLANNING_SCHEMA.STEP'] * TYPEOF(SELF)) = 1)) OR (
       SIZEOF(QUERY ( pdr <*
       get_property_definition_representations(SELF) | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) >= 0);
END_ENTITY; -- feature_definition

ENTITY feature_dependency
 SUBTYPE OF (group_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM' IN
       TYPEOF(SELF.related_group);
  wr2: 'PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM' IN
       TYPEOF(SELF.relating_group);
END_ENTITY; -- feature_dependency

ENTITY feature_identification_item
 SUBTYPE OF (class);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( adr <* QUERY ( ga <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.GROUP_ASSIGNMENT.ASSIGNED_GROUP')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT' IN
       TYPEOF(ga)) ) | (SIZEOF(QUERY ( ins <* adr.items | (
       'PROCESS_PLANNING_SCHEMA.INSTANCED_FEATURE' IN TYPEOF(ins)) ))
       = 1) )) = 0);
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.GROUP_RELATIONSHIP.RELATING_GROUP')
       | ('PROCESS_PLANNING_SCHEMA.FEATURE_INTERACTION' IN TYPEOF(
       adr)) )) >= 0;
  wr3: NOT (SIZEOF(QUERY ( adr <* QUERY ( ga <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.GROUP_ASSIGNMENT.ASSIGNED_GROUP')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT' IN
       TYPEOF(ga)) ) | (SIZEOF(QUERY ( ins <* adr.items | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_PROCESS' IN TYPEOF(ins)) ))
       <= 1) )) = 0);
END_ENTITY; -- feature_identification_item

ENTITY feature_interaction
 SUBTYPE OF (group_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM' IN
       TYPEOF(SELF.related_group);
  wr2: 'PROCESS_PLANNING_SCHEMA.FEATURE_IDENTIFICATION_ITEM' IN
       TYPEOF(SELF.relating_group);
END_ENTITY; -- feature_interaction

ENTITY feature_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
```

```
        + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
        used_representation.items | (NOT (
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(srwp_i))) )) >
        0)) )) > 0) )) = 0;
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
        name = 'base feature placement')) )) > 1)) )) = 0)) )) = 0;
END_ENTITY; -- feature_pattern

ENTITY feature_process
 SUBTYPE OF (action_method);
 WHERE
  wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')
        | (('PROCESS_PLANNING_SCHEMA.SEQUENTIAL_METHOD' IN TYPEOF(
        sar)) AND ('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY'
        IN TYPEOF(sar.related_method))) )) >= 1;
  wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')
        | (('PROCESS_PLANNING_SCHEMA.SERIAL_ACTION_METHOD' IN
        TYPEOF(sar)) AND (
        'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(
        sar.related_method))) )) = 1;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
        NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | ((
        SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
        TYPEOF(it)) = 1) AND (it.name = 'process parameter')) )) >=
        0)) )) = 0)) )) = 0;
END_ENTITY; -- feature_process

ENTITY fillet
 SUBTYPE OF (transition_feature);
 WHERE
  wr1: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
```

 

```
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
      = 0)) )) = 0);
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'fillet face')) )) = 1)) )) = 0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
```

```
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
        'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'second face shape')) )) = 1)) ))
        = 0;
 END_ENTITY; -- fillet

 ENTITY fixture_assembly
  SUBTYPE OF (action_resource);
  WHERE
   wr1: SIZEOF(QUERY ( adr <* SELF.usage | ((
        'PROCESS_PLANNING_SCHEMA.ACTION_METHOD' IN TYPEOF(adr)) AND
        (adr.name = 'assembly instruction')) )) = 1;
   wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (adr
        .description = 'configuration') )) = 1;
   wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATING_RESOURCE')
        | (('PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_RELATIONSHIP'
        IN TYPEOF(ap)) AND (SIZEOF([
        'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
        'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT'] *
        TYPEOF(ap.related_resource)) = 1)) )) >= 1;
   wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT.ITEMS')
        | ('PROCESS_PLANNING_SCHEMA.FIXTURE_CONTRACT' IN TYPEOF(adr
        .assigned_contract)) )) >= 1;
   wr5: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
        | ('PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(adr
        .assigned_document)) )) <= 1;
   wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
        | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
        TYPEOF(ar)) ) | (NOT (SIZEOF(QUERY ( pdp_imp <*
        QUERY ( pdp <* arr.operations | (
        'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS' IN
                        TYPEOF(pdp))   )   |   (SIZEOF(QUERY   (   ppa   <*
USEDIN(pdp_imp,'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
        | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
        property_or_shape)) AND (ppa.description =
        'fixture assembly shape')) )) = 0) )) = 0)) )) >= 0;
 END_ENTITY; -- fixture_assembly

 ENTITY fixture_assembly_element
  SUBTYPE OF (action_resource);
  WHERE
   wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
        | (('PROCESS_PLANNING_SCHEMA.DOCUMENT' IN TYPEOF(adr.
        assigned_document)) AND (adr.assigned_document.description =
```

 

```
        'fixture catalogue number')) )) <= 1;
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
        | (adr.assigned_organization.description =
        'fixture company name') )) = 1;
  wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (adr
        .description = 'fixture element identification') )) = 1;
  wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
        NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | (NOT (SIZEOF(QUERY ( rpr <* apr.representation.items | ((
        'PROCESS_PLANNING_SCHEMA.MASS_MEASURE_WITH_UNIT' IN TYPEOF(
        rpr)) AND (rpr.name = 'fixture element weight')) )) <= 1)) ))
        = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
        | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
        TYPEOF(ar)) ) | (NOT (SIZEOF(QUERY ( pdp_imp <*
        QUERY ( pdp <* arr.operations | (
        'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS' IN
        TYPEOF(pdp)) ) | (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
       'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
        | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
        property_or_shape)) AND (ppa.description =
        'fixture assembly element shape')) )) = 0) )) = 0)) )) >= 0;
END_ENTITY; -- fixture_assembly_element

ENTITY fixture_assembly_relationship
 SUBTYPE OF (action_resource_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(SELF.
        relating_resource);
  wr2: SIZEOF(['PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
        'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT'] *
        TYPEOF(SELF.related_resource)) = 1;
END_ENTITY; -- fixture_assembly_relationship

ENTITY fixture_contract
 SUBTYPE OF (contract);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')
       | (adr.assigned_action_request.description = 'design order') ))
        = 1;
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')
        | (adr.assigned_action_request.description =
        'fabrication order') )) = 1;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')
        | (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
        IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (
       'PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION' IN TYPEOF(it)) ))
        <= 1))) )) = 0;
END_ENTITY; -- fixture_contract

ENTITY fixture_machine_mounting
 SUBTYPE OF (fixture_setup);
END_ENTITY; -- fixture_machine_mounting
```

```
ENTITY fixture_machine_unmounting
 SUBTYPE OF (fixture_setup);
END_ENTITY; -- fixture_machine_unmounting

ENTITY fixture_pallet_mounting
 SUBTYPE OF (fixture_setup);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 0) ))
       = 0);
END_ENTITY; -- fixture_pallet_mounting

ENTITY fixture_pallet_unmounting
 SUBTYPE OF (fixture_setup);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 0) ))
       = 0);
END_ENTITY; -- fixture_pallet_unmounting

ENTITY fixture_setup
 SUBTYPE OF (process_plan_activity);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       'PROCESS_PLANNING_SCHEMA.MOUNTING_POSITION' IN TYPEOF(ap)) ))
       <= 1;
  wr2: (NOT type_check_function(SELF,[
       'PROCESS_PLANNING_SCHEMA.FIXTURE_PALLET_MOUNTING',
       'PROCESS_PLANNING_SCHEMA.FIXTURE_PALLET_UNMOUNTING'],1)) OR
       (NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 1) ))
       = 0));
  wr3: NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.PALLET' IN TYPEOF(it)) )) >= 0) ))
       = 0);
  wr4: NOT (SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(it)) ))
       >= 1) )) = 0);
END_ENTITY; -- fixture_setup
```

 

```
ENTITY flat_face
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'course of travel occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'path feature component usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (((
       'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(
       sdr.relating_shape_aspect)) AND (sdr.relating_shape_aspect.
      description = 'linear')) AND (sdr.name = 'course of travel')) ))
       = 1)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'removal boundary occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE'
       IN TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
       'removal boundary')) )) = 1)) )) = 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'enclosed boundary occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
       relating_shape_aspect)) = 1) AND (sdr.relating_shape_aspect.
       description = 'boundary')) )) = 1)) )) <= 1)) )) = 0;
  wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
```

```
       SELF) | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) >= 0;
  wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
       'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
       csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
       QUERY ( sar <* csa.component_relationships | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) AND (SIZEOF(['PROCESS_PLANNING_SCHEMA.BOSS',
       'PROCESS_PLANNING_SCHEMA.PROTRUSION'] * TYPEOF(sar.
       related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'removal depth')) )) <= 1)) ))
       = 0)) )) = 0;
END_ENTITY; -- flat_face

ENTITY flatness_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1: NOT (('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- flatness_tolerance

ENTITY founded_item;
END_ENTITY; -- founded_item

ENTITY functionally_defined_transformation;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- functionally_defined_transformation

ENTITY gear
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
 'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ((('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

 

```
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN( pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
```

```
        SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((
        SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
  'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
        IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
        QUERY ( it <* impl_rep.used_representation.items | ((
        SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
     | ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
```

```
       QUERY ( it <* impl_rep.used_representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
        | ((sar.description = 'applied shape') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
       IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
        0;
END_ENTITY; -- gear

ENTITY general_property;
  id     : identifier;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- general_property

ENTITY general_property_association;
  name        : label;
  description    : OPTIONAL text;
  base_definition  : general_property;
  derived_definition : derived_property_select;
 WHERE
  wr1: SIZEOF(USEDIN(derived_definition,('PROCESS_PLANNING_SCHEMA.' +
       'GENERAL_PROPERTY_ASSOCIATION.') + 'DERIVED_DEFINITION')) =
       1;
  wr2: derived_definition.name = base_definition.name;
END_ENTITY; -- general_property_association

ENTITY generic_manufacturing_resource
 SUPERTYPE OF (ONEOF (resource_with_representation,
   resource_with_material))
 SUBTYPE OF (action_resource);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( rpr <* apr.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(rpr)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(rpr\measure_with_unit.value_component))) AND (rpr.
       name = 'resource quantity')) )) >= 1) )) = 0)) )) = 0);
  wr2: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( rpr <* apr.representation.items | ((((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(rpr)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(rpr\measure_with_unit.value_component))) AND (rpr.
       name = 'resource quantity')) AND (rpr\measure_with_unit.
```

 

```
       unit_component.name = 'each')) )) = 1) )) = 0)) )) = 0);
END_ENTITY; -- generic_manufacturing_resource

ENTITY geometric_alignment
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY; -- geometric_alignment

ENTITY geometric_curve_set
 SUBTYPE OF (geometric_set);
 WHERE
  wr1: SIZEOF(QUERY ( temp <* SELF\geometric_set.elements | (
       'PROCESS_PLANNING_SCHEMA.SURFACE' IN TYPEOF(temp)) )) = 0;
END_ENTITY; -- geometric_curve_set

ENTITY geometric_intersection
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) > 1;
END_ENTITY; -- geometric_intersection

ENTITY geometric_representation_context
 SUBTYPE OF (representation_context);
  coordinate_space_dimension : dimension_count;
END_ENTITY; -- geometric_representation_context

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,
   shell_based_surface_model,shell_based_wireframe_model,
   edge_based_wireframe_model,geometric_set))
 SUBTYPE OF (representation_item);
 DERIVE
  dim : dimension_count := dimension_of(SELF);
 WHERE
  wr1: SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (NOT
       ('PROCESS_PLANNING_SCHEMA.GEOMETRIC_REPRESENTATION_CONTEXT'
       IN TYPEOF(using_rep.context_of_items))) )) = 0;
END_ENTITY; -- geometric_representation_item

ENTITY geometric_set
 SUPERTYPE OF (geometric_curve_set)
 SUBTYPE OF (geometric_representation_item);
  elements : SET [1:?] OF geometric_set_select;
END_ENTITY; -- geometric_set

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

ENTITY geometric_tolerance_relationship;
  name             : label;
  description         : text;
  relating_geometric_tolerance : geometric_tolerance;
```

```
  related_geometric_tolerance : geometric_tolerance;
END_ENTITY; -- geometric_tolerance_relationship

ENTITY geometric_tolerance_with_datum_reference
 SUBTYPE OF (geometric_tolerance);
  datum_system : SET [1:?] OF datum_reference;
END_ENTITY; -- geometric_tolerance_with_datum_reference

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

ENTITY geometrically_bounded_surface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       > 0;
  wr3: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
       NOT ((('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION') IN
       TYPEOF(mi\mapped_item.mapping_source.mapped_representation))
       AND (SIZEOF(QUERY ( mr_it <* mi\mapped_item.mapping_source.
       mapped_representation.items | (
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET' IN TYPEOF(mr_it)) ))
       > 0))) )) = 0;
  wr4: SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET' IN TYPEOF(it)) ) | (
       NOT (SIZEOF(QUERY ( pnt <* QUERY ( gsel <* gs\geometric_set.
      elements | ('PROCESS_PLANNING_SCHEMA.POINT' IN TYPEOF(gsel)) )
       | (NOT gbsf_check_point(pnt)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET' IN TYPEOF(it)) ) | (
       NOT (SIZEOF(QUERY ( cv <* QUERY ( gsel <* gs\geometric_set.
      elements | ('PROCESS_PLANNING_SCHEMA.CURVE' IN TYPEOF(gsel)) )
       | (NOT gbsf_check_curve(cv)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET' IN TYPEOF(it)) ) | (
       NOT (SIZEOF(QUERY ( sf <* QUERY ( gsel <* gs\geometric_set.
     elements | ('PROCESS_PLANNING_SCHEMA.SURFACE' IN TYPEOF(gsel)) )
       | (NOT gbsf_check_surface(sf)) )) = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( gs <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_SET' IN TYPEOF(it)) ) | (
       SIZEOF(QUERY ( gsel <* gs\geometric_set.elements | (
       'PROCESS_PLANNING_SCHEMA.SURFACE' IN TYPEOF(gsel)) )) > 0) ))
       > 0;
END_ENTITY; -- geometrically_bounded_surface_shape_representation

ENTITY geometrically_bounded_wireframe_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
```

 

```
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF(TYPEOF(it) * [
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM']) = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF(TYPEOF(it) * [
       'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM']) = 1) )) >= 1;
  wr3: SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET' IN TYPEOF(it)) )
       | (NOT (SIZEOF(QUERY ( crv <* QUERY ( elem <* gcs\
       geometric_set.elements | ('PROCESS_PLANNING_SCHEMA.CURVE' IN
       TYPEOF(elem)) ) | (NOT
       valid_geometrically_bounded_wf_curve(crv)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET' IN TYPEOF(it)) )
       | (NOT (SIZEOF(QUERY ( pnts <* QUERY ( elem <* gcs\
       geometric_set.elements | ('PROCESS_PLANNING_SCHEMA.POINT' IN
       TYPEOF(elem)) ) | (NOT
       valid_geometrically_bounded_wf_point(pnts)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET' IN TYPEOF(it)) )
       | (NOT (SIZEOF(QUERY ( cnc <* QUERY ( elem <* gcs\
       geometric_set.elements | ('PROCESS_PLANNING_SCHEMA.CONIC' IN
       TYPEOF(elem)) ) | (NOT (
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D' IN TYPEOF(cnc\
       conic.position))) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( gcs <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.GEOMETRIC_CURVE_SET' IN TYPEOF(it)) )
       | (NOT (SIZEOF(QUERY ( pline <* QUERY ( elem <* gcs\
       geometric_set.elements | ('PROCESS_PLANNING_SCHEMA.POLYLINE'
     IN TYPEOF(elem)) ) | (NOT (SIZEOF(pline\polyline.points) > 2)) ))
       = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
       NOT (('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION') IN
      TYPEOF(mi\mapped_item.mapping_source.mapped_representation))) ))
       = 0;
END_ENTITY; -- geometrically_bounded_wireframe_shape_representation

ENTITY global_uncertainty_assigned_context
 SUBTYPE OF (representation_context);
  uncertainty : SET [1:?] OF uncertainty_measure_with_unit;
END_ENTITY; -- global_uncertainty_assigned_context

ENTITY global_unit_assigned_context
 SUBTYPE OF (representation_context);
  units : SET [1:?] OF unit;
END_ENTITY; -- global_unit_assigned_context

ENTITY group;
  name    : label;
  description : OPTIONAL text;
 DERIVE
  id : identifier := get_id_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- group

ENTITY group_assignment
```

```
 ABSTRACT SUPERTYPE;
  assigned_group : group;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- group_assignment

ENTITY group_relationship;
  name      : label;
  description  : OPTIONAL text;
  relating_group : group;
  related_group : group;
END_ENTITY; -- group_relationship

ENTITY hole_bottom
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(USEDIN(pd,'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0);
  wr5 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
       USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
```

 

```
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | ((sar.description =
      'hole bottom usage') AND (
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATING_SHAPE_ASPECT') | ((sar.description =
      'hole bottom usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
      = 'bottom condition occurrence') AND (
      'PROCESS_PLANNING_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
      related_shape_aspect.of_shape.definition))) AND (
      'PROCESS_PLANNING_SCHEMA.HOLE_BOTTOM' IN TYPEOF(fcr.
      relating_shape_aspect))) )) >= 1;
  wr15: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'taper diameter')) )) = 1)) ))
      = 0)) )) = 0);
```

 

```
END_ENTITY; -- hole_bottom

ENTITY hyperbola
 SUBTYPE OF (conic);
  semi_axis   : positive_length_measure;
  semi_imag_axis : positive_length_measure;
END_ENTITY; -- hyperbola

ENTITY id_attribute;
  attribute_value : identifier;
  identified_item : id_attribute_select;
END_ENTITY; -- id_attribute

ENTITY identification_assignment
 ABSTRACT SUPERTYPE;
  assigned_id : identifier;
  role    : identification_role;
END_ENTITY; -- identification_assignment

ENTITY identification_role;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- identification_role

ENTITY illustration
 SUBTYPE OF (document);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (NOT (SIZEOF(QUERY ( d <* adr.items | (
      'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) ))
       >= 1)) )) = 0;
  wr2: SIZEOF(QUERY ( aduc <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.VIEW_REFERENCE' IN TYPEOF(aduc\
     document_usage_constraint_assignment.assigned_document_usage)) ))
       >= 0;
END_ENTITY; -- illustration

ENTITY in_facility_location
 SUBTYPE OF (resource_property);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* ap.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'building or area')) )) = 1) ))
       = 0);
  wr2: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* ap.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'location code')) )) = 1) )) = 0);
  wr3: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* ap.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'sublocation')) )) = 1) )) = 0);
END_ENTITY; -- in_facility_location
```

```
ENTITY instanced_feature
 SUBTYPE OF (feature_definition, shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(SELF.
       of_shape.definition);
  wr2: SELF.product_definitional;
END_ENTITY; -- instanced_feature

ENTITY intersection_curve
 SUBTYPE OF (surface_curve);
 WHERE
  wr1: SIZEOF(SELF\surface_curve.associated_geometry) = 2;
  wr2: associated_surface(SELF\surface_curve.associated_geometry[1])
       <> associated_surface(SELF\surface_curve.associated_geometry
       [2]);
END_ENTITY; -- intersection_curve

ENTITY known_source
 SUBTYPE OF (external_source, pre_defined_item);
END_ENTITY; -- known_source

ENTITY length_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.LENGTH_UNIT' IN TYPEOF(SELF\
       measure_with_unit.unit_component);
END_ENTITY; -- length_measure_with_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

ENTITY library_class_version_assignment
 SUBTYPE OF (applied_external_identification_assignment);
END_ENTITY; -- library_class_version_assignment

ENTITY library_property_version_assignment
 SUBTYPE OF (applied_external_identification_assignment);
END_ENTITY; -- library_property_version_assignment

ENTITY limits_and_fits;
  form_variance : label;
  zone_variance : label;
  grade     : label;
  source    : text;
END_ENTITY; -- limits_and_fits

ENTITY line
 SUBTYPE OF (curve);
  pnt : cartesian_point;
  dir : vector;
 WHERE
```

 

```
  wr1: dir.dim = pnt.dim;
END_ENTITY; -- line

ENTITY line_profile_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1: (NOT (('PROCESS_PLANNING_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,'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | (sar.
       name IN ['affected plane association',
       'resulting intersection curve association']) )) = 1;
END_ENTITY; -- line_profile_tolerance

ENTITY linear_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
```

```
       TYPEOF(it)) = 2) AND (it.name = 'profile length')) )) = 1)) ))
       = 0)) )) = 0;
END_ENTITY; -- linear_profile

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

ENTITY location_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(SELF.items) = 1;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
       'PROCESS_PLANNING_SCHEMA.POINT' IN TYPEOF(it))) )) = 0;
END_ENTITY; -- location_shape_representation

ENTITY loop
 SUPERTYPE OF (ONEOF (vertex_loop,edge_loop,poly_loop))
 SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

ENTITY machine
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.ORGANIZATION' IN TYPEOF(adr.
       assigned_organization)) AND (adr.assigned_organization.
       description = 'machine company name')) )) = 1;
  wr2: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')
       | ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
       IN TYPEOF(arr)) ) | ('PROCESS_PLANNING_SCHEMA.CONTROLLER' IN
       TYPEOF(ar2.relating_resource)) )) <= 1;
  wr3: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')
       | ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
       IN TYPEOF(arr)) ) | ('PROCESS_PLANNING_SCHEMA.WORKSTATION'
       IN TYPEOF(ar2.relating_resource)) )) = 1;
  wr4: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
 'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATED_RESOURCE')
       | ('PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP'
       IN TYPEOF(arr)) ) | ('PROCESS_PLANNING_SCHEMA.PALLET' IN
       TYPEOF(ar2.relating_resource)) )) = 1;
END_ENTITY; -- machine

ENTITY machine_element_relationship
 SUBTYPE OF (action_resource_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(SELF.
       related_resource);
  wr2: SIZEOF(['PROCESS_PLANNING_SCHEMA.CONTROLLER',
       'PROCESS_PLANNING_SCHEMA.PALLET',
       'PROCESS_PLANNING_SCHEMA.WORKSTATION'] * TYPEOF(SELF.
       relating_resource)) = 1;
END_ENTITY; -- machine_element_relationship
```

 

```
ENTITY machine_setup
 SUBTYPE OF (process_plan_activity);
 WHERE
  wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (NOT (SIZEOF(QUERY ( it <* am.resources |
       ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) >= 1)) ))
       = 0;
END_ENTITY; -- machine_setup

ENTITY machine_usage
 SUBTYPE OF (requirement_for_action_resource);
 WHERE
  wr1 : SIZEOF(QUERY ( w <* SELF.resources | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(w)) )) = 1;
  wr2 : SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
        | ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
        TYPEOF(adr.assigned_document)) )) <= 1;
  wr3 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'maximum spindle speed')) ))
       <= 1)) )) = 0)) )) = 0);
  wr4 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'maximum feedrate')) )) <=
       1)) )) = 0)) )) = 0);
  wr5 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
        TYPEOF(it\measure_with_unit.value_component))) AND (it.
       name = 'number of control axis')) )) <= 1)) )) = 0)) )) = 0);
 wr6: (NOT(SIZEOF(QUERY ( rp <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
      'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')|
```

```
      ('PROCESS_PLANNING_SCHEMA.REPRESENTATION'
      IN TYPEOF(pdr.representation))) |
      ((impl_rep.representation.name = 'machine usage parameter')AND
      (SIZEOF(QUERY ( it <* impl_rep.representation.items |
     ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND  (it.name  IN
    ['x-axis table size','y-axis table size'])
    ) ))<=2))  ))=0))))=0));
  wr7 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND (it.description IN ['true','false',
       'unknown'])) AND (it.name = 'table indexing function')) ))
       <= 1)) )) = 0)) )) = 0);
  wr8 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'positioning accuracy')) ))
       <= 1)) )) = 0)) )) = 0);
  wr9 : NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'axis range of motion ')) ))
       >= 0)) )) = 0)) )) = 0);
  wr10: NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND (it.name =
       'axis range of motion description')) )) >= 0)) )) = 0)) ))
       = 0);
  wr11: NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
```

 

```
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
        TYPEOF(it\measure_with_unit.value_component))) AND (it.
       name = 'number of simultaneous axis')) )) <= 1)) )) = 0)) ))
       = 0);
  wr12: NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
        | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'machine usage parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'spindle power')) )) <= 1)) ))
       = 0)) )) = 0);
END_ENTITY; -- machine_usage

ENTITY machining_process
 SUBTYPE OF (manufacturing_process);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) )) >= 0;
END_ENTITY; -- machining_process

ENTITY machining_tolerance
 SUBTYPE OF (action_property);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'inside tolerance')) )) = 1) ))
       = 0);
  wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'outside tolerance')) )) = 1) ))
       = 0);
  wr3: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'total tolerance')) )) <= 1) ))
       = 0);
END_ENTITY; -- machining_tolerance
```

```
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

ENTITY manifold_solid_brep
 SUBTYPE OF (solid_model);
  outer : closed_shell;
END_ENTITY; -- manifold_solid_brep

ENTITY manifold_surface_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2 : SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       > 0;
  wr3 : SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
    NOT (('PROCESS_PLANNING_SCHEMA.MANIFOLD_SURFACE_SHAPE_REPRESENTATION'
       IN TYPEOF(mi\mapped_item.mapping_source.
       mapped_representation)) AND (SIZEOF(QUERY ( mr_it <* mi\
       mapped_item.mapping_source.mapped_representation.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
       TYPEOF(mr_it)) )) > 0))) )) = 0;
  wr4 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( sh <* sbsm\
       shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.OPEN_SHELL',
       'PROCESS_PLANNING_SCHEMA.ORIENTED_CLOSED_SHELL',
       'PROCESS_PLANNING_SCHEMA.CLOSED_SHELL'] * TYPEOF(sh)) = 1)) ))
       = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
       shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
       QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT (
       'PROCESS_PLANNING_SCHEMA.FACE_SURFACE' IN TYPEOF(fa))) )) =
        0)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
       shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
       QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
       'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR
       msf_surface_check(fa\face_surface.face_geometry))) )) = 0)) ))
       = 0)) )) = 0;
  wr7 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
```

 

```
      shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
      QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
      'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
      SIZEOF(QUERY ( bnds <* fa.bounds | (NOT (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.EDGE_LOOP',
      'PROCESS_PLANNING_SCHEMA.VERTEX_LOOP'] * TYPEOF(bnds.bound))
      = 1)) )) = 0))) )) = 0)) )) = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
      shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
      QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
      'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
      SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
      'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
       | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
      NOT ('PROCESS_PLANNING_SCHEMA.EDGE_CURVE' IN TYPEOF(oe.
      edge_element))) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
      shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
      QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
      'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
      SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
      'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
       | (NOT (SIZEOF(QUERY ( oe_cv <* QUERY ( oe <* elp_fbnds\
      path.edge_list | ('PROCESS_PLANNING_SCHEMA.EDGE_CURVE' IN
      TYPEOF(oe.edge_element)) ) | (NOT (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE',
      'PROCESS_PLANNING_SCHEMA.CONIC',
      'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA',
      'PROCESS_PLANNING_SCHEMA.LINE',
      'PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D',
      'PROCESS_PLANNING_SCHEMA.PCURVE',
      'PROCESS_PLANNING_SCHEMA.POLYLINE',
      'PROCESS_PLANNING_SCHEMA.SURFACE_CURVE'] * TYPEOF(oe_cv.
      edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) )) =
       0))) )) = 0)) )) = 0)) )) = 0;
wr10: SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
      shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
      QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
      'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
      SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
      'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
       | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
      NOT msf_curve_check(oe.edge_element\edge_curve.
      edge_geometry)) )) = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0;
wr11: SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
      shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
      QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
      'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
      SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
      'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
       | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
      NOT (('PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(oe.
      edge_element.edge_start)) AND (
```

```
        'PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(oe.
        edge_element.edge_end)))) )) = 0)) )) = 0))) )) = 0)) )) =
        0)) )) = 0;
  wr12: SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
        'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
        TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
        'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
        SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* fa.bounds | (
        'PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN TYPEOF(bnds.bound)) )
         | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds\path.edge_list | (
        NOT ((SIZEOF(['PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT',
        'PROCESS_PLANNING_SCHEMA.DEGENERATE_PCURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_CURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_SURFACE'] * TYPEOF(oe.
        edge_element.edge_start\vertex_point.vertex_geometry)) = 1)
        AND (SIZEOF(['PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT',
        'PROCESS_PLANNING_SCHEMA.DEGENERATE_PCURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_CURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_SURFACE'] * TYPEOF(oe.
        edge_element.edge_end\vertex_point.vertex_geometry)) = 1))) ))
        = 0)) )) = 0))) )) = 0)) )) = 0)) )) = 0;
  wr13: SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
        'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
        TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
        'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
        SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
       'PROCESS_PLANNING_SCHEMA.VERTEX_LOOP' IN TYPEOF(bnds.bound)) )
         | (NOT ('PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(
        vlp_fbnds\vertex_loop.loop_vertex))) )) = 0))) )) = 0)) ))
        = 0)) )) = 0;
  wr14: SIZEOF(QUERY ( sbsm <* QUERY ( it <* SELF.items | (
        'PROCESS_PLANNING_SCHEMA.SHELL_BASED_SURFACE_MODEL' IN
        TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( cfs <* sbsm\
        shell_based_surface_model.sbsm_boundary | (NOT (SIZEOF(
        QUERY ( fa <* cfs\connected_face_set.cfs_faces | (NOT ((
        'PROCESS_PLANNING_SCHEMA.ADVANCED_FACE' IN TYPEOF(fa)) OR (
        SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* fa.bounds | (
       'PROCESS_PLANNING_SCHEMA.VERTEX_LOOP' IN TYPEOF(bnds.bound)) )
         | (NOT (SIZEOF(['PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT',
        'PROCESS_PLANNING_SCHEMA.DEGENERATE_PCURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_CURVE',
        'PROCESS_PLANNING_SCHEMA.POINT_ON_SURFACE'] * TYPEOF(
        vlp_fbnds\vertex_loop.loop_vertex\vertex_point.
        vertex_geometry)) = 1)) )) = 0))) )) = 0)) )) = 0)) )) = 0;
END_ENTITY; -- manifold_surface_shape_representation

ENTITY manufacturing_activity
 SUPERTYPE OF (ONEOF (ancillary_activity,part_routing,part_unmounting,
   process_activity,validation,part_mounting))
 SUBTYPE OF (process_plan_activity);
 WHERE
wr1: (NOT ('PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
        IN TYPEOF(SELF))) OR (SIZEOF(QUERY ( adr <* SELF\
        action_method_with_associated_documents.documents | (
       'PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM' IN TYPEOF(adr)) ))
        <= 1);
  wr2: SIZEOF(QUERY ( sar <* USEDIN(SELF,
```

 

```
'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')
       | ((
       'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
       IN TYPEOF(sar)) AND (NOT (SIZEOF(QUERY ( edi <* sar.items |
       ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_SCHEMA' IN
       TYPEOF(edi)) )) <= 1))) )) = 0;
END_ENTITY; -- manufacturing_activity

ENTITY manufacturing_activity_relationship
 SUBTYPE OF (sequential_method);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       relating_method);
  wr2: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY' IN TYPEOF(SELF
       .related_method);
END_ENTITY; -- manufacturing_activity_relationship

ENTITY manufacturing_process
 SUBTYPE OF (action_method);
 WHERE
  wr1: SIZEOF(QUERY ( amr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')
       | (('PROCESS_PLANNING_SCHEMA.SINGLE_ACTIVITY_RELATIONSHIP'
       IN TYPEOF(amr)) AND (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MACHINE_SETUP',
       'PROCESS_PLANNING_SCHEMA.TOOL_SETUP',
       'PROCESS_PLANNING_SCHEMA.FIXTURE_SETUP',
       'PROCESS_PLANNING_SCHEMA.ANCILLARY_SETUP'] * TYPEOF(amr.
       related_method)) = 1)) )) >= 1;
  wr2: SELF.purpose IN ['legacy nc','non nc','integrated nc'];
  wr3: SIZEOF(QUERY ( cm <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | (NOT (
       SIZEOF(QUERY ( ppa <* USEDIN(cm,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(ppa.property_or_shape)) AND (ppa.name =
       'assigned feature')) )) = 1)) )) >= 0;
  wr4: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')
      | (('PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY_RELATIONSHIP'
       IN TYPEOF(sar)) AND (
       'PROCESS_PLANNING_SCHEMA.MANUFACTURING_ACTIVITY' IN TYPEOF(
       sar.related_method))) )) >= 1;
  wr5: SIZEOF(QUERY ( cm <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | (NOT (
       SIZEOF(QUERY ( ppa <* USEDIN(cm,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP' IN
       TYPEOF(ppa.property_or_shape)) AND (ppa.description =
       'intermediate shape')) )) >= 0)) )) = 0;
  wr6: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')|
    ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE'
     IN TYPEOF(arr)) ) |
     (NOT (SIZEOF(QUERY ( rp <* USEDIN(am,
     'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') |
     ('PROCESS_PLANNING_SCHEMA.PART_HOLDING_POSITION' IN TYPEOF(rp)) ))
     = 1)) )) = 0;
END_ENTITY; -- manufacturing_process

ENTITY manufacturing_process_relationship
```

```
 SUBTYPE OF (sequential_method);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       related_method);
  wr2: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       relating_method);
END_ENTITY; -- manufacturing_process_relationship

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

ENTITY marking
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
       + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) )) = 0)) ))
       = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (it.name = 'special instructions')) )) <= 1)) ))
       = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
```

 

```
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'font name')) )) <= 1)) )) = 0)) ))
        = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'character height')) )) <= 1)) ))
        = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'character spacing')) )) <=
       1)) )) = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'applied shape') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN
       TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
END_ENTITY; -- marking

ENTITY mass_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.MASS_UNIT' IN TYPEOF(SELF\
       measure_with_unit.unit_component);
END_ENTITY; -- mass_measure_with_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

ENTITY material_designation;
  name    : label;
  definitions : SET [1:?] OF characterized_definition;
END_ENTITY; -- material_designation

ENTITY material_property
 SUBTYPE OF (property_definition);
 UNIQUE
  ur1 : name, definition;
 WHERE
  wr1: ('PROCESS_PLANNING_SCHEMA.CHARACTERIZED_OBJECT' IN TYPEOF(SELF\
       property_definition.definition)) OR (SIZEOF(bag_to_set(
       USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
       QUERY ( temp <* bag_to_set(USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'MATERIAL_PROPERTY_REPRESENTATION') IN TYPEOF(temp)) )) = 0);
END_ENTITY; -- material_property

ENTITY material_property_representation
 SUBTYPE OF (property_definition_representation);
  dependent_environment : data_environment;
END_ENTITY; -- material_property_representation

ENTITY measure_qualification;
  name       : label;
  description    : text;
  qualified_measure : measure_with_unit;
  qualifiers    : SET [1:?] OF value_qualifier;
 WHERE
  wr1: SIZEOF(QUERY ( temp <* qualifiers | (
     'PROCESS_PLANNING_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp)) ))
       < 2;
END_ENTITY; -- measure_qualification

ENTITY measure_representation_item
 SUBTYPE OF (representation_item, measure_with_unit);
END_ENTITY; -- measure_representation_item

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

ENTITY modified_geometric_tolerance
 SUBTYPE OF (geometric_tolerance);
  modifier : limit_condition;
END_ENTITY; -- modified_geometric_tolerance

ENTITY modified_pattern
 SUBTYPE OF (shape_aspect);
 WHERE
```

 

```
  wr1: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) ) | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.REPLICATE_FEATURE',
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) ) | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN',
       'PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(fcr.
       related_shape_aspect.of_shape.definition)) = 1) AND (fcr.
       description = 'base pattern')) )) = 1;
  wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (SIZEOF(QUERY ( msar <* USEDIN(
       sar.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | (((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.PATTERN_OFFSET_MEMBERSHIP',
       'PROCESS_PLANNING_SCHEMA.PATTERN_OMIT_MEMBERSHIP'] * TYPEOF(
       sar)) = 1) AND (sar.description = 'modified pattern')) AND (
       sar :<>: msar)) )) >= 1) )) = 0;
END_ENTITY; -- modified_pattern

ENTITY mounting_position
 SUBTYPE OF (action_property);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1) )) = 0);
  wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(it)) AND
       (it.name = 'location origin')) )) = 1) )) = 0);
  wr3: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLANE' IN TYPEOF(it)) AND (it.name
       = 'reference plane')) )) = 1) )) = 0);
END_ENTITY; -- mounting_position

ENTITY name_attribute;
  attribute_value : label;
  named_item   : name_attribute_select;
END_ENTITY; -- name_attribute

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
```

```
ENTITY next_assembly_usage_occurrence
 SUBTYPE OF (assembly_component_usage);
END_ENTITY; -- next_assembly_usage_occurrence

ENTITY ngon_closed_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
       + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) <= 1)) ))
      = 0)) )) = 0;
END_ENTITY; -- ngon_closed_profile

ENTITY ngon_shape_representation
 SUBTYPE OF (shape_representation_with_parameters);
 WHERE
  wr1: SIZEOF(SELF.items) = 5;
  wr2: SIZEOF(QUERY ( it <* SELF.items | ((
      'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
      name = 'orientation')) )) = 1;
  wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
  wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1;
  wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
      'diameter across flats'])) )) = 1;
  wr6: SIZEOF(QUERY ( it <* SELF.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
      TYPEOF(it\measure_with_unit.value_component))) AND (it.name
      = 'number of sides')) )) = 1;
END_ENTITY; -- ngon_shape_representation

ENTITY non_machining_process
 SUBTYPE OF (manufacturing_process);
END_ENTITY; -- non_machining_process
```

```
ENTITY object_role;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- object_role

ENTITY offset_curve_3d
 SUBTYPE OF (curve);
  basis_curve  : curve;
  distance    : length_measure;
  self_intersect : LOGICAL;
  ref_direction : direction;
 WHERE
  wr1: (basis_curve.dim = 3) AND (ref_direction.dim = 3);
END_ENTITY; -- offset_curve_3d

ENTITY offset_surface
 SUBTYPE OF (surface);
  basis_surface : surface;
  distance    : length_measure;
  self_intersect : LOGICAL;
END_ENTITY; -- offset_surface

ENTITY open_path_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
       'PROCESS_PLANNING_SCHEMA.PATH_SHAPE_REPRESENTATION' IN
```

 

```
        TYPEOF(pdr.used_representation)) )) = 1)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'profile limit')) )) <= 1)) )) =
        0;
END_ENTITY; -- open_path_profile

ENTITY open_shell
 SUBTYPE OF (connected_face_set);
END_ENTITY; -- open_shell

ENTITY organization;
  id     : OPTIONAL identifier;
  name    : label;
  description : OPTIONAL text;
END_ENTITY; -- organization

ENTITY organization_assignment
 ABSTRACT SUPERTYPE;
  assigned_organization : organization;
  role         : organization_role;
END_ENTITY; -- organization_assignment

ENTITY organization_role;
  name : label;
 DERIVE
  description : text := get_description_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- organization_role

ENTITY organizational_address
 SUBTYPE OF (address);
  organizations : SET [1:?] OF organization;
  description  : OPTIONAL text;
END_ENTITY; -- organizational_address

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,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- organizational_project

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 ('PROCESS_PLANNING_SCHEMA.ORIENTED_CLOSED_SHELL' IN TYPEOF(
       SELF.closed_shell_element));
END_ENTITY; -- oriented_closed_shell

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 ('PROCESS_PLANNING_SCHEMA.ORIENTED_EDGE' IN TYPEOF(SELF.
       edge_element));
END_ENTITY; -- oriented_edge

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 ('PROCESS_PLANNING_SCHEMA.ORIENTED_FACE' IN TYPEOF(SELF.
       face_element));
END_ENTITY; -- oriented_face

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 ('PROCESS_PLANNING_SCHEMA.ORIENTED_OPEN_SHELL' IN TYPEOF(
       SELF.open_shell_element));
END_ENTITY; -- oriented_open_shell

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 ('PROCESS_PLANNING_SCHEMA.ORIENTED_PATH' IN TYPEOF(SELF.
       path_element));
END_ENTITY; -- oriented_path
```

 

```
ENTITY oriented_surface
 SUBTYPE OF (surface);
  orientation : BOOLEAN;
END_ENTITY; -- oriented_surface

ENTITY outer_boundary_curve
 SUBTYPE OF (boundary_curve);
END_ENTITY; -- outer_boundary_curve

ENTITY outer_round
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: (NOT (SELF\characterized_object.description = 'outer diameter'))
       OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
       + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
       + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
       = 0);
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'v-shape boundary occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (('PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'reduced size occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
      AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((('PROCESS_PLANNING_SCHEMA.TAPER' IN
      TYPEOF(sdr.relating_shape_aspect)) AND (
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'feature length')) )) <= 1)) ))
```

 

```
      = 0)) )) = 0);
END_ENTITY; -- outer_round

ENTITY outside_profile
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
       description = 'boundary occurrence') )) = 1)) )) = 0)) OR (
       SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (('PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
       SIZEOF(['PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
       'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE',
       'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
       'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
       'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
       'PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE',
       'PROCESS_PLANNING_SCHEMA.VEE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
       relating_shape_aspect)) = 1) AND (sdr.relating_shape_aspect
       .description = 'outside boundary')) )) = 1) )) = 1)) )) = 0);
  wr4 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (NOT (
      SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile floor usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ((
      'PROCESS_PLANNING_SCHEMA.PROFILE_FLOOR' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (
      'PROCESS_PLANNING_SCHEMA.OUTSIDE_PROFILE' IN TYPEOF(sdr.
     related_shape_aspect.of_shape.definition))) )) = 1)) )) = 0)) ))
      = 0);
  wr5 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND ((
      'PROCESS_PLANNING_SCHEMA.' + 'SHAPE_DEFINING_RELATIONSHIP')
      IN TYPEOF(sar))) ) | (((SIZEOF([
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
      description = 'complex boundary occurrence') )) = 1)) )) =
      0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
```

 

```
      TYPEOF(sar))) ) | (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE',
      'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
      'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.VEE_PROFILE',
      'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
      'PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE',
      'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
      relating_shape_aspect)) = 1) )) = 1) )) = 1)) )) = 0);
  wr7 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
      description = 'partial circular boundary occurrence') )) =
      1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr8 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
     description = 'closed circular boundary occurrence') )) = 1)) ))
      = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr9 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
      description = 'open rectangular boundary occurrence') )) =
      1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(sdr.
      relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr10: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (sa_occ.
      description = 'closed rectangular boundary occurrence') ))
      = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
  wr11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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) | ((
      'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'maximum feature limit')) )) >=
      0;
END_ENTITY; -- outside_profile

ENTITY pallet
 SUBTYPE OF (action_resource);
```

 

```
END_ENTITY; -- pallet

ENTITY pallet_machine_mounting
 SUBTYPE OF (part_mounting);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) )) = 1;
  wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.PALLET' IN TYPEOF(ap)) )) = 1;
END_ENTITY; -- pallet_machine_mounting

ENTITY parabola
 SUBTYPE OF (conic);
  focal_dist : length_measure;
 WHERE
  wr1: focal_dist <> 0;
END_ENTITY; -- parabola

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

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

ENTITY parametric_representation_context
 SUBTYPE OF (representation_context);
END_ENTITY; -- parametric_representation_context

ENTITY part_contract
 SUBTYPE OF (contract);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.ORGANIZATION' IN TYPEOF(adr.
       assigned_organization)) AND (adr.assigned_organization.
       description = 'binding aggrement')) )) = 1;
  wr2: SIZEOF(QUERY ( ca <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
       IN TYPEOF(ca)) AND (NOT (SIZEOF(QUERY ( pdf <* ca.items | ((
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_FORMATION' IN
       TYPEOF(pdf)) AND (NOT (SIZEOF(QUERY ( pd <* USEDIN(pdf,
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION.FORMATION') | (
       pd.frame_of_reference.name = 'part definition') )) = 1))) ))
       = 1))) )) = 0;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
       IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | ((
       'PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE' IN TYPEOF(it)) AND (
```

```
       it.description = 'contract document')) )) <= 1))) )) = 0;
  wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.ORGANIZATION' IN TYPEOF(adr.
       assigned_organization)) AND (SIZEOF(QUERY ( op <* USEDIN(adr
       .assigned_organization,
'PROCESS_PLANNING_SCHEMA.ORGANIZATIONAL_PROJECT.RESPONSIBLE_ORGANIZATIONS')
       | (op.description = 'project number') )) = 1)) )) = 1;
END_ENTITY; -- part_contract

ENTITY part_fixture_mounting
 SUBTYPE OF (part_mounting);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(ap)) ))
       >= 1;
END_ENTITY; -- part_fixture_mounting

ENTITY part_holding_position
 SUBTYPE OF (resource_property);
 WHERE
  wr1: SIZEOF(QUERY ( pdp_imp <* QUERY ( pdp <* SELF.resource.
       operations | ('PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN
       TYPEOF(pdp)) ) | (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
       property_or_shape)) AND (ppa.description =
       'part holding mating shape')) )) = 0) )) >= 0;
  wr2: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( ap <* apr.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(ap)) AND (ap.name = 'part holding position type'))
       AND (ap.description IN ['jack','locator','clamp'])) )) = 1) ))
       = 0);
  wr3: ('PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(SELF.resource)) AND (SIZEOF(QUERY ( rfar <* SELF.
       resource.resources | (
       'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(rfar)) ))
       = 1);
  wr4: NOT (SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( ap <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(ap)) AND
       (ap.name = 'part holding location')) )) <= 1) )) = 0);
END_ENTITY; -- part_holding_position

ENTITY part_machine_mounting
 SUBTYPE OF (part_mounting);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(ap)) )) = 1;
END_ENTITY; -- part_machine_mounting

ENTITY part_mounting
 SUPERTYPE OF (ONEOF (part_fixture_mounting,part_machine_mounting,
   pallet_machine_mounting))
 SUBTYPE OF (manufacturing_activity);
 WHERE
```

 

```
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(it)) AND
       (it.name = 'part location origin')) )) = 1) )) = 1) )) = 1;
  wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       'PROCESS_PLANNING_SCHEMA.MOUNTING_POSITION' IN TYPEOF(ap)) ))
       <= 1;
  wr3: SIZEOF(QUERY ( cm <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | (NOT (
       SIZEOF(QUERY ( ppa <* USEDIN(cm,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(ppa.property_or_shape)) AND (ppa.description =
       'part mounting part shape')) )) <= 1)) )) = 0;
END_ENTITY; -- part_mounting

ENTITY part_routing
 SUBTYPE OF (manufacturing_activity);
END_ENTITY; -- part_routing

ENTITY part_unmounting
 SUBTYPE OF (manufacturing_activity);
 WHERE
  wr1: SELF.name IN ['pallet machine unmounting',
       'part fixture unmounting','part machine unmounting'];
END_ENTITY; -- part_unmounting

ENTITY partial_circular_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
```

```
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
        name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
        = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
        = 0)) )) = 0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
        TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'profile limit')) )) <= 1)) )) =
        0;
END_ENTITY; -- partial_circular_profile

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

ENTITY path_feature_component
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

 

```
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0);
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
      name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr5 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
      'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
  wr13: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
      'PROCESS_PLANNING_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

ENTITY path_shape_representation
```

 

```
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(SELF.items) >= 1;
  wr2: SIZEOF(QUERY ( i <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE',
       'PROCESS_PLANNING_SCHEMA.EDGE_CURVE',
       'PROCESS_PLANNING_SCHEMA.PATH'] * TYPEOF(i)) = 1) )) >= 1;
END_ENTITY; -- path_shape_representation

ENTITY pattern_offset_membership
 SUBTYPE OF (feature_component_relationship);
 WHERE
  wr1 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
       relating_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
       QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.
       related_shape_aspect.of_shape,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN',
       'PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(pdr
       .definition)) = 1) )) = 0) )) = 0;
  wr2 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
       related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
       'modified pattern') AND (
       'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
       relating_shape_aspect))) )) >= 1;
  wr3 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
       related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
       'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
       relating_shape_aspect)) AND (NOT (SIZEOF(QUERY ( modfcr <*
       QUERY ( modsar <* USEDIN(fcr.relating_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
       SIZEOF(['PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN',
       'PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(
       modsar.related_shape_aspect.of_shape.definition)) = 1) AND
       (modsar :<>: fcr)) ) | (NOT (modfcr.related_shape_aspect.
       of_shape.definition :=: SELF.relating_shape_aspect.of_shape
       .definition)) )) = 0))) )) = 0;
  wr4 : (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
       SELF.relating_shape_aspect.of_shape.definition))) OR (
       SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(USEDIN(pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 2)) ))
       = 0);
  wr5 : (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(
       SELF.relating_shape_aspect.of_shape.definition))) OR (
```

```
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(USEDIN(pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
      = 0);
  wr6 : (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
      SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
  wr7 : (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(it\measure_with_unit.value_component))) AND (it.
      name = 'index number')) )) = 1)) )) = 0)) )) = 0);
  wr8 : (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT' IN
      TYPEOF(it)) AND (it.name = 'offset')) )) = 1)) )) = 0)) ))
      = 0);
  wr9 : (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(impl_rep.
      used_representation.items) = 3)) )) = 0)) )) = 0);
  wr10: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
      SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
```

 

```
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
         TYPEOF(it\measure_with_unit.value_component))) AND (it.
        name = 'row index')) )) = 1)) )) = 0)) )) = 0);
   wr11: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
        SELF.relating_shape_aspect.of_shape.definition))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
         TYPEOF(it\measure_with_unit.value_component))) AND (it.
        name = 'column index')) )) = 1)) )) = 0)) )) = 0);
   wr12: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
        SELF.relating_shape_aspect.of_shape.definition))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT' IN
        TYPEOF(it)) AND (it.name = 'offset distance')) )) = 1)) ))
        = 0)) )) = 0);
   wr13: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
        SELF.relating_shape_aspect.of_shape.definition))) OR (
        SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
         TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'offset direction')) )) = 1)) ))
        = 0);
END_ENTITY; -- pattern_offset_membership

ENTITY pattern_omit_membership
 SUBTYPE OF (feature_component_relationship);
 WHERE
  wr1: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
        relating_shape_aspect,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATING_SHAPE_ASPECT') | ((
        'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
        QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.related_shape_aspect
        .of_shape,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
```

```
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN',
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(pdr.
      definition)) = 1) )) = 0) )) = 0;
  wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
      related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
      'modified pattern') AND (
      'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
      relating_shape_aspect))) )) >= 1;
  wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
      related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
      'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN' IN TYPEOF(fcr.
      relating_shape_aspect)) AND (NOT (SIZEOF(QUERY ( modfcr <*
      QUERY ( modsar <* USEDIN(fcr.relating_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
      SIZEOF(['PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN',
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN'] * TYPEOF(
      modsar.related_shape_aspect.of_shape.definition)) = 1) AND (
      modsar :<>: fcr)) ) | (NOT (modfcr.related_shape_aspect.
      of_shape.definition :=: SELF.relating_shape_aspect.of_shape.
      definition)) )) = 0))) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(USEDIN(pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) )) =
      0;
  wr5: (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF
      .relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
      SIZEOF(pdr.used_representation.items) = 1)) )) = 0)) )) = 0);
  wr6: (NOT ('PROCESS_PLANNING_SCHEMA.CIRCULAR_PATTERN' IN TYPEOF(SELF
      .relating_shape_aspect.of_shape.definition))) OR (SIZEOF(
      QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
      TYPEOF(it\measure_with_unit.value_component))) AND (it.name
      = 'index number')) )) = 1)) )) = 0)) )) = 0);
  wr7: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
      SELF.relating_shape_aspect.of_shape.definition))) OR (
```

 

```
       SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
       SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
  wr8: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
       SELF.relating_shape_aspect.of_shape.definition))) OR (
       SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(it\measure_with_unit.value_component))) AND (it.name
       = 'row index')) )) = 1)) )) = 0)) )) = 0);
  wr9: (NOT ('PROCESS_PLANNING_SCHEMA.RECTANGULAR_PATTERN' IN TYPEOF(
       SELF.relating_shape_aspect.of_shape.definition))) OR (
       SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(it\measure_with_unit.value_component))) AND (it.name
       = 'column index')) )) = 1)) )) = 0)) )) = 0);
END_ENTITY; -- pattern_omit_membership

ENTITY pcurve
 SUBTYPE OF (curve);
  basis_surface   : surface;
  reference_to_curve : definitional_representation;
 WHERE
  wr1: SIZEOF(reference_to_curve\representation.items) = 1;
  wr2: 'PROCESS_PLANNING_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

ENTITY perpendicular_to
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- perpendicular_to

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

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

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,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization

ENTITY personal_address
 SUBTYPE OF (address);
  people   : SET [1:?] OF person;
  description : OPTIONAL text;
END_ENTITY; -- personal_address

ENTITY placement
 SUPERTYPE OF (ONEOF (axis1_placement,axis2_placement_2d,
   axis2_placement_3d))
 SUBTYPE OF (geometric_representation_item);
  location : cartesian_point;
END_ENTITY; -- placement

ENTITY planar_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(SELF.items) = 1;
  wr2: SIZEOF(QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.PLANE' IN TYPEOF(it)) )) = 1;
END_ENTITY; -- planar_shape_representation

ENTITY plane
 SUBTYPE OF (elementary_surface);
END_ENTITY; -- plane

ENTITY plane_angle_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_UNIT' IN TYPEOF(SELF\
       measure_with_unit.unit_component);
END_ENTITY; -- plane_angle_measure_with_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

ENTITY plus_minus_tolerance;
  range        : tolerance_method_definition;
  toleranced_dimension : dimensional_characteristic;
 UNIQUE
  ur1 : toleranced_dimension;
END_ENTITY; -- plus_minus_tolerance

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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
       .description = 'pocket depth occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | (((sar.description =
       'path feature component usage') AND (sar.name =
       'pocket depth')) AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ((
       'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(
       sdr.relating_shape_aspect)) AND (sdr.relating_shape_aspect.
       description = 'linear')) )) = 1)) )) = 1)) )) = 0;
  wr3 : SIZEOF(QUERY ( pdr <* get_property_definition_representations(
       SELF) |
     ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF(pdr.used_representation)) )) = 1;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'boundary occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (SIZEOF([
      'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
      'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE',
      'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
      'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.VEE_PROFILE',
      'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'closed boundary occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (
      'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
  wr9 : (NOT (SELF\characterized_object.description =
      'open rectangular')) OR (SIZEOF(QUERY ( pds <*
      QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

 

```
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
     .description = 'open boundary occurrence') AND (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATED_SHAPE_ASPECT') | ((sar.description =
     'profile usage') AND (
     'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
     TYPEOF(sar))) ) | (
     'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE' IN TYPEOF(sdr.
     relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr10: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
     .description = 'bottom condition occurrence') AND (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATED_SHAPE_ASPECT') | ((sar.description =
     'pocket bottom usage') AND (
     'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((
     'PROCESS_PLANNING_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr.
     relating_shape_aspect)) AND (
     'PROCESS_PLANNING_SCHEMA.POCKET' IN TYPEOF(sdr.
   related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
     = 0;
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
     .description = 'change in boundary occurrence') AND (
     SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATING_SHAPE_ASPECT') | ((sar.description =
     'taper usage') AND (
     'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((('PROCESS_PLANNING_SCHEMA.TAPER' IN
     TYPEOF(fcr.relating_shape_aspect)) AND (
     'PROCESS_PLANNING_SCHEMA.POCKET' IN TYPEOF(fcr.
     related_shape_aspect.of_shape.definition))) AND (fcr.
     related_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
     .description = 'enclosed boundary occurrence') AND (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATED_SHAPE_ASPECT') | ((sar.description =
     'profile usage') AND (
     'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
     TYPEOF(sar))) ) | (
     'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'bottom condition occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'pocket bottom usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (((
      'PROCESS_PLANNING_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
      .description = 'bottom condition occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'pocket bottom usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (((
      'PROCESS_PLANNING_SCHEMA.POCKET_BOTTOM' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (
      'PROCESS_PLANNING_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) | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
      'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
      csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
      QUERY ( sar <* csa.component_relationships | ((
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) AND (SIZEOF(['PROCESS_PLANNING_SCHEMA.BOSS',
      'PROCESS_PLANNING_SCHEMA.PROTRUSION'] * TYPEOF(sar.
      related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1);
END_ENTITY; -- pocket
```

 

```
ENTITY pocket_bottom
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) = 0)) ))
       = 0);
  wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | ((sar.description =
       'pocket bottom usage') AND (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
       = 'bottom condition occurrence') AND (
       'PROCESS_PLANNING_SCHEMA.POCKET' IN TYPEOF(fcr.
       related_shape_aspect.of_shape.definition))) AND (
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | ((sar.description =
       'pocket bottom usage') AND (
       'PROCESS_PLANNING_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

ENTITY point
 SUPERTYPE OF (ONEOF (cartesian_point,point_on_curve,point_on_surface,
   point_replica,degenerate_pcurve))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- point

ENTITY point_on_curve
 SUBTYPE OF (point);
  basis_curve   : curve;
  point_parameter : parameter_value;
END_ENTITY; -- point_on_curve

ENTITY point_on_surface
 SUBTYPE OF (point);
  basis_surface   : surface;
  point_parameter_u : parameter_value;
  point_parameter_v : parameter_value;
END_ENTITY; -- point_on_surface

ENTITY point_replica
 SUBTYPE OF (point);
  parent_pt   : point;
  transformation : cartesian_transformation_operator;
```

 

```
 WHERE
  wr1: transformation.dim = parent_pt.dim;
  wr2: acyclic_point_replica(SELF,parent_pt);
END_ENTITY; -- point_replica

ENTITY poly_loop
 SUBTYPE OF (loop, geometric_representation_item);
  polygon : LIST [3:?] OF UNIQUE cartesian_point;
END_ENTITY; -- poly_loop

ENTITY polyline
 SUBTYPE OF (bounded_curve);
  points : LIST [2:?] OF cartesian_point;
END_ENTITY; -- polyline

ENTITY position_tolerance
 SUBTYPE OF (geometric_tolerance_with_datum_reference);
 WHERE
  wr1: SIZEOF(SELF\geometric_tolerance_with_datum_reference.
       datum_system) <= 3;
END_ENTITY; -- position_tolerance

ENTITY pre_defined_item;
  name : label;
END_ENTITY; -- pre_defined_item

ENTITY precision_qualifier;
  precision_value : INTEGER;
END_ENTITY; -- precision_qualifier

ENTITY process_activity
 SUBTYPE OF (manufacturing_activity);
 WHERE
  wr1: SIZEOF(QUERY ( sar <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_ASSIGNMENT.ASSIGNED_ACTION_METHOD')|
     (('PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_METHOD_ASSIGNMENT'
       IN TYPEOF(sar)) AND (NOT (SIZEOF(QUERY ( edi <* sar.items |
       ('PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION' IN
       TYPEOF(edi)) )) >= 0))) )) = 0;
  wr2: SIZEOF(QUERY ( cm <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | ((
       'PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(cm))
       AND (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | ((ppa.description = 'process activity part shape') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
       property_or_shape))) )) <= 1))) )) = 0;
  wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
  'PROCESS_PLANNING_SCHEMA.ACTION_METHOD_RELATIONSHIP.RELATING_METHOD')
      | (('PROCESS_PLANNING_SCHEMA.DEFINING_ACTION_METHOD_RELATIONSHIP'
       IN TYPEOF(sar)) AND ('PROCESS_PLANNING_SCHEMA.MACHINE_SETUP'
       IN TYPEOF(sar.related_method))) )) <= 1;
  wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(ap)) ))
       <= 1;
  wr5: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(ap)) )) >=
       1;
  wr6: SIZEOF(QUERY ( ap <* USEDIN(SELF,
```

```
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
      'PROCESS_PLANNING_SCHEMA.MACHINING_TOLERANCE' IN TYPEOF(ap)) ))
       <= 1;
END_ENTITY; -- process_activity

ENTITY process_plan_activity
 SUPERTYPE OF (ONEOF (ancillary_setup,fixture_setup,machine_setup,
   manufacturing_activity,tool_setup))
 SUBTYPE OF (action_method);
 WHERE
  wr1 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (ap.name
       = 'frequency') )) <= 1;
  wr2 : SIZEOF(QUERY ( adr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
        | (('PROCESS_PLANNING_SCHEMA.ORGANIZATION' IN TYPEOF(adr.
       assigned_organization)) AND (adr.role.name =
       'activity organization id')) )) = 1;
  wr3 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND (it.name = 'special instruction')) )) >=
        0)) )) = 0)) )) = 0;
  wr4 : (NOT
    ('PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
       IN TYPEOF(SELF))) OR (SIZEOF(QUERY ( adr <* SELF\
       action_method_with_associated_documents.documents | ((
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
       TYPEOF(adr)) AND (adr.kind.product_data_type =
       'supplemental documents')) )) >= 0);
  wr5 : (NOT
    ('PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
       IN TYPEOF(SELF))) OR (SIZEOF(QUERY ( adr <* SELF\
       action_method_with_associated_documents.documents | (
       'PROCESS_PLANNING_SCHEMA.ILLUSTRATION' IN TYPEOF(adr)) ))
       >= 0);
  wr6 : (NOT
    ('PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
       IN TYPEOF(SELF))) OR (SIZEOF(QUERY ( adr <* SELF\
       action_method_with_associated_documents.documents | (
       'PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(adr)) ))
       >= 0);
  wr7 : (NOT
    ('PROCESS_PLANNING_SCHEMA.ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS'
       IN TYPEOF(SELF))) OR (SIZEOF(QUERY ( edi <* SELF\
       action_method_with_associated_documents.documents | (
       'PROCESS_PLANNING_SCHEMA.VIEW_REFERENCE' IN TYPEOF(edi)) ))
       >= 0);
  wr8 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       SIZEOF(['PROCESS_PLANNING_SCHEMA.PRODUCTION_RATE',
       'PROCESS_PLANNING_SCHEMA.ALLOWED_TIME'] * TYPEOF(ap)) = 1) ))
       >= 1;
  wr9 : SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE.USAGE') | (
       'PROCESS_PLANNING_SCHEMA.GENERIC_MANUFACTURING_RESOURCE' IN
        TYPEOF(ap)) )) >= 0;
  wr10: SIZEOF(QUERY ( cm <* USEDIN(SELF,
```

 

```
        'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | ((
        'PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(cm))
        AND (NOT (SIZEOF(QUERY ( ppa <* USEDIN(cm,
      'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
         | ((ppa.description = 'process plan activity material')
        AND ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION' IN
        TYPEOF(ppa.property_or_shape.definition))) )) <= 1))) )) =
        0;
  wr11: (NOT (('PROCESS_PLANNING_SCHEMA.' +
        'ACTION_METHOD_WITH_ASSOCIATED_DOCUMENTS') IN TYPEOF(SELF)))
        OR (SIZEOF(SELF\action_method_with_associated_documents.
        documents) >= 0);
  wr12: SIZEOF(QUERY ( ap <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
        NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
         | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | ((
        ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
         TYPEOF(it\measure_with_unit.value_component))) AND (it.
        name = 'activity number')) )) >= 0)) )) = 0)) )) = 0;
END_ENTITY; -- process_plan_activity

ENTITY process_plan_security
 SUBTYPE OF (security_classification);
 WHERE
  wr1: SIZEOF(QUERY ( da <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') | (
        (('PROCESS_PLANNING_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da))
        AND ('PROCESS_PLANNING_SCHEMA.CALENDAR_DATE' IN TYPEOF(da.
       assigned_date))) AND (da.role.name = 'classification date')) ))
        = 1;
  wr2: SIZEOF(QUERY ( da <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.APPLIED_DATE_ASSIGNMENT.ITEMS') | (
        (('PROCESS_PLANNING_SCHEMA.DATE_ASSIGNMENT' IN TYPEOF(da))
        AND ('PROCESS_PLANNING_SCHEMA.CALENDAR_DATE' IN TYPEOF(da.
      assigned_date))) AND (da.role.name = 'declassification date')) ))
        = 1;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SECURITY_CLASSIFICATION_ASSIGNMENT.'
        + 'ASSIGNED_SECURITY_CLASSIFICATION') |
  (('PROCESS_PLANNING_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
        IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (
      'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(it)) ))
        = 1))) )) = 0;
  wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
  'SECURITY_CLASSIFICATION_ASSIGNMENT.ASSIGNED_SECURITY_CLASSIFICATION')
   | (('PROCESS_PLANNING_SCHEMA.APPLIED_SECURITY_CLASSIFICATION_ASSIGNMENT'
        IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(it)) ))
        = 1))) )) = 0;
END_ENTITY; -- process_plan_security

ENTITY process_plan_specification
 SUBTYPE OF (document);
 WHERE
  wr1: SIZEOF(QUERY ( duc <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')
        | (duc.subject_element = 'revision') )) = 1;
  wr2: SIZEOF(QUERY ( duc <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE')
```

```
       | (duc.subject_element = 'subclass') )) = 1;
END_ENTITY; -- process_plan_specification

ENTITY process_plan_version
 SUBTYPE OF (product_definition_process);
 WHERE
  wr1: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (ap.
       description = 'manufacturing gt code') )) <= 1;
  wr2: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(it\measure_with_unit.value_component))) AND (it.name
       = 'quantity of parts')) )) >= 0)) )) = 0)) )) = 0;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (it.name = 'special instruction')) )) >= 0)) ))
       = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       NOT (SIZEOF(QUERY ( apr <* USEDIN(ap,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (NOT (SIZEOF(QUERY ( it <* apr.representation.items | (it
       .name = 'auxilary header') )) >= 0)) )) = 0)) )) = 0;
  wr5: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY.DEFINITION') | (
       'PROCESS_PLANNING_SCHEMA.RANGE_OF_PARTS' IN TYPEOF(ap)) ))
       <= 1;
  wr6: SIZEOF(QUERY ( ppp <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROCESS_PRODUCT_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(
       ppp.defined_product)) AND (ppp.defined_product.name =
       'material')) )) <= 1;
  wr7: SIZEOF(['PROCESS_PLANNING_SCHEMA.NON_MACHINING_PROCESS',
       'PROCESS_PLANNING_SCHEMA.MACHINING_PROCESS'] * TYPEOF(SELF.
       chosen_method)) = 1;
  wr8: SIZEOF(['PROCESS_PLANNING_SCHEMA.NON_MACHINING_PROCESS',
       'PROCESS_PLANNING_SCHEMA.MACHINING_PROCESS'] * TYPEOF(SELF.
       chosen_method)) = 1;
  wr9: SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_ASSIGNMENT.ASSIGNED_ACTION')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_ASSIGNMENT' IN
       TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_DEPENDENCY' IN TYPEOF(it)) ))
       <= 1))) )) = 0;
END_ENTITY; -- process_plan_version

ENTITY process_product_association;
  name      : label;
  description   : text;
  defined_product : characterized_product_definition;
  process     : product_definition_process;
END_ENTITY; -- process_product_association
```

 

```
ENTITY process_property_association;
  name       : label;
  description    : text;
  process      : property_process;
  property_or_shape : property_or_shape_select;
END_ENTITY; -- process_property_association

ENTITY product;
  id         : identifier;
  name        : label;
  description    : OPTIONAL text;
  frame_of_reference : SET [1:?] OF product_context;
END_ENTITY; -- product

ENTITY product_category;
  name    : label;
  description : OPTIONAL text;
 DERIVE
  id : identifier := get_id_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- product_category

ENTITY product_context
 SUBTYPE OF (application_context_element);
  discipline_type : label;
END_ENTITY; -- product_context

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,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- product_definition

ENTITY product_definition_context
 SUBTYPE OF (application_context_element);
  life_cycle_stage : label;
END_ENTITY; -- product_definition_context

ENTITY product_definition_effectivity
 SUBTYPE OF (effectivity);
  usage : product_definition_relationship;
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'EFFECTIVITY_ASSIGNMENT.ASSIGNED_EFFECTIVITY')) = 0;
END_ENTITY; -- product_definition_effectivity

ENTITY product_definition_formation;
  id     : identifier;
  description : OPTIONAL text;
  of_product : product;
 UNIQUE
  ur1 : id, of_product;
END_ENTITY; -- product_definition_formation
```

```
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

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

ENTITY product_definition_shape
 SUBTYPE OF (property_definition);
 UNIQUE
  ur1 : definition;
 WHERE
  wr1: SIZEOF([
       'PROCESS_PLANNING_SCHEMA.CHARACTERIZED_PRODUCT_DEFINITION',
       'PROCESS_PLANNING_SCHEMA.CHARACTERIZED_OBJECT'] * TYPEOF(
       SELF\property_definition.definition)) > 0;
END_ENTITY; -- product_definition_shape

ENTITY product_definition_usage
 SUPERTYPE OF (ONEOF (make_from_usage_option,assembly_component_usage))
 SUBTYPE OF (product_definition_relationship);
 UNIQUE
  ur1 : id, relating_product_definition, related_product_definition;
 WHERE
  wr1: acyclic_product_definition_relationship(SELF,[SELF\
       product_definition_relationship.related_product_definition],
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_USAGE');
END_ENTITY; -- product_definition_usage

ENTITY product_definition_with_associated_documents
 SUBTYPE OF (product_definition);
  documentation_ids : SET [1:?] OF document;
END_ENTITY; -- product_definition_with_associated_documents

ENTITY product_related_product_category
 SUBTYPE OF (product_category);
  products : SET [1:?] OF product;
END_ENTITY; -- product_related_product_category

ENTITY production_rate
 SUBTYPE OF (action_property);
 WHERE
  wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.TIME_MEASURE_WITH_UNIT'] * TYPEOF(
       it)) = 2) AND (it.name = 'time per unit')) )) = 1) )) = 1;
  wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( it <* apr.representation.items | ((
```

 

```
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'unit quantity')) )) = 1) )) = 1;
   wr3: SIZEOF(QUERY ( apr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
        | (SIZEOF(QUERY ( it <* apr.representation.items | ((
        'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND (it.name = 'production rate source')) ) =
        1) )) = 1;
  wr4: (NOT(SIZEOF(QUERY ( apr <* USEDIN(SELF,
         'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')|
         (SIZEOF(QUERY ( it <* apr.representation.items |
         (('PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
         IN TYPEOF(it)) AND (it.name = 'allowed type'))))<=1)))=0));
END_ENTITY; -- production_rate

ENTITY profile_floor
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
```

```
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
       = 0);
  wr7 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | ((sar.description =
       'profile floor usage') AND (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | ((
       'PROCESS_PLANNING_SCHEMA.OUTSIDE_PROFILE' IN TYPEOF(fcr.
       related_shape_aspect.of_shape.definition)) AND (
       'PROCESS_PLANNING_SCHEMA.PROFILE_FLOOR' IN TYPEOF(fcr.
       relating_shape_aspect))) )) >= 1;
  wr8 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
       QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'floor')) )) = 1)) )) = 1);
END_ENTITY; -- profile_floor

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

ENTITY property_definition;
  name    : label;
  description : OPTIONAL text;
  definition : characterized_definition;
 DERIVE
```

 

```
  id : identifier := get_id_value(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- property_definition

ENTITY property_definition_relationship;
  name             : label;
  description         : text;
  relating_property_definition : property_definition;
  related_property_definition : property_definition;
END_ENTITY; -- property_definition_relationship

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,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- property_definition_representation

ENTITY property_process
 SUBTYPE OF (action);
  identification : identifier;
 INVERSE
  properties : SET [1:?] OF process_property_association FOR process;
END_ENTITY; -- property_process

ENTITY protrusion
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'shape volume occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'volume shape usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (sdr.relating_shape_aspect.description =
       'volume shape') )) = 1)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
       SELF) | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
```

```
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) >= 0;
END_ENTITY; -- protrusion

ENTITY qualified_representation_item
 SUBTYPE OF (representation_item);
  qualifiers : SET [1:?] OF value_qualifier;
 WHERE
  wr1: SIZEOF(QUERY ( temp <* qualifiers | (
     'PROCESS_PLANNING_SCHEMA.PRECISION_QUALIFIER' IN TYPEOF(temp)) ))
       < 2;
END_ENTITY; -- qualified_representation_item

ENTITY quasi_uniform_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY; -- quasi_uniform_curve

ENTITY quasi_uniform_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY; -- quasi_uniform_surface

ENTITY range_of_parts
 SUBTYPE OF (action_property);
 WHERE
  wr1: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( rep <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(rep)) AND (rep.name = 'low value')) )) = 1) )) = 1;
  wr2: SIZEOF(QUERY ( apr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.ACTION_PROPERTY_REPRESENTATION.PROPERTY')
       | (SIZEOF(QUERY ( rep <* apr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(rep)) AND (rep.name = 'high value')) )) = 1) )) = 1;
END_ENTITY; -- ran ge_of_parts

ENTITY ratio_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.RATIO_UNIT' IN TYPEOF(SELF\
       measure_with_unit.unit_component);
END_ENTITY; -- ratio_measure_with_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

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

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

ENTITY rectangular_closed_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
       USEDIN(pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
      name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
      = 0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
      = 0;
  wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) <= 1)) ))
      = 0)) )) = 0;
END_ENTITY; -- rectangular_closed_profile

ENTITY rectangular_composite_surface
 SUBTYPE OF (bounded_surface);
  segments : LIST [1:?] OF LIST [1:?] OF surface_patch;
 DERIVE
  n_u : INTEGER := SIZEOF(segments);
  n_v : INTEGER := SIZEOF(segments[1]);
 WHERE
  wr1: SIZEOF(QUERY ( s <* segments | (n_v <> SIZEOF(s)) )) = 0;
  wr2: constraints_rectangular_composite_surface(SELF);
END_ENTITY; -- rectangular_composite_surface

ENTITY rectangular_pattern
 SUBTYPE OF (replicate_feature);
 WHERE
  wr1 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

 

```
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATING_SHAPE_ASPECT') | (
     'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar)) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
     IN TYPEOF(sdr.related_shape_aspect)) )) = 1) )) <= 5)) )) =
      0;
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
     'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) )) = 1)) )) = 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | (((
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
     TYPEOF(it)) AND ('PROCESS_PLANNING_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```

```
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | (((
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
        TYPEOF(it)) AND ('PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'row spacing')) )) = 1)) ))
        = 0)) )) = 0;
  wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
        'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'column spacing')) )) = 1)) ))
        = 0)) )) = 0;
  wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
         | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
        pd,'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
        impl_rep.used_representation.items | ((
        'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
        name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
END_ENTITY; -- rectangular_pattern

ENTITY rectangular_trimmed_surface
 SUBTYPE OF (bounded_surface);
  basis_surface : surface;
  u1      : parameter_value;
  u2      : parameter_value;
  v1      : parameter_value;
  v2      : parameter_value;
  usense    : BOOLEAN;
  vsense    : BOOLEAN;
 WHERE
  wr1: u1 <> u2;
  wr2: v1 <> v2;
```

 

```
  wr3: ((('PROCESS_PLANNING_SCHEMA.ELEMENTARY_SURFACE' IN TYPEOF(
       basis_surface)) AND (NOT ('PROCESS_PLANNING_SCHEMA.PLANE' IN
       TYPEOF(basis_surface)))) OR (
       'PROCESS_PLANNING_SCHEMA.SURFACE_OF_REVOLUTION' IN TYPEOF(
       basis_surface))) OR (usense = (u2 > u1));
  wr4: (('PROCESS_PLANNING_SCHEMA.SPHERICAL_SURFACE' IN TYPEOF(
       basis_surface)) OR (
       'PROCESS_PLANNING_SCHEMA.TOROIDAL_SURFACE' IN TYPEOF(
       basis_surface))) OR (vsense = (v2 > v1));
END_ENTITY; -- rectangular_trimmed_surface

ENTITY referenced_modified_datum
 SUBTYPE OF (datum_reference);
  modifier : limit_condition;
END_ENTITY; -- referenced_modified_datum

ENTITY relationship_condition;
  name           : label;
  applicable_relationships : SET [1:?] OF relationship_with_condition;
  condition_description  : text;
END_ENTITY; -- relationship_condition

ENTITY removal_volume
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'shape volume occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'volume shape usage') AND (('PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (sdr.
      relating_shape_aspect.description = 'volume shape') )) = 1)) ))
       = 1)) )) = 0;
END_ENTITY; -- removal_volume

ENTITY reparametrised_composite_curve_segment
 SUBTYPE OF (composite_curve_segment);
  param_length : parameter_value;
 WHERE
  wr1: param_length > 0;
END_ENTITY; -- reparametrised_composite_curve_segment

ENTITY replicate_feature
 SUPERTYPE OF (ONEOF (circular_pattern,rectangular_pattern,
   feature_pattern))
 SUBTYPE OF (feature_definition);
 WHERE
```

```
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar)) ) | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.REPLICATE_FEATURE',
       'PROCESS_PLANNING_SCHEMA.INSTANCED_FEATURE'] * TYPEOF(fcr.
     related_shape_aspect)) >= 1) AND (fcr.name = 'pattern basis')) ))
       = 1;
  wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | (NOT (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) )) + SIZEOF(QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | (NOT (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ))) = 0;
END_ENTITY; -- replicate_feature

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,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
  wr2: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- representation

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

ENTITY representation_item;
  name : label;
 WHERE
  wr1: SIZEOF(using_representations(SELF)) > 0;
END_ENTITY; -- representation_item

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

ENTITY requirement_for_action_resource
 SUBTYPE OF (action_resource_requirement);
  resources : SET [1:?] OF action_resource;
END_ENTITY; -- requirement_for_action_resource

ENTITY resource_property;
  name    : label;
  description : text;
  resource  : characterized_resource_definition;
END_ENTITY; -- resource_property

ENTITY resource_property_representation;
  name      : label;
  description  : text;
  property    : resource_property;
  representation : representation;
END_ENTITY; -- resource_property_representation

ENTITY resource_requirement_type;
  name    : label;
  description : text;
END_ENTITY; -- resource_requirement_type

ENTITY resource_with_material
 SUBTYPE OF (generic_manufacturing_resource);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
       | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
       TYPEOF(ar)) ) | (NOT (SIZEOF(QUERY ( pdp_imp <*
       QUERY ( pdp <* arr.operations | (
       'PROCESS_PLANNING_SCHEMA.PROPERTY_PROCESS' IN TYPEOF(pdp)) )
       | (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | ((('PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION' IN
       TYPEOF(ppa.property_or_shape)) AND (ppa.description =
       'resource material')) AND (
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION' IN TYPEOF(ppa.
       property_or_shape.definition))) )) = 1) )) = 0)) )) = 0);
END_ENTITY; -- resource_with_material

ENTITY resource_with_representation
 SUBTYPE OF (generic_manufacturing_resource);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       SIZEOF(QUERY ( rpr <* USEDIN(ap,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
| ('PROCESS_PLANNING_SCHEMA.EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF(rpr.representation)) )) = 1) )) = 0);
END_ENTITY; -- resource_with_representation

ENTITY revision
 SUBTYPE OF (action_relationship);
 WHERE
  wr1: SIZEOF(QUERY ( rc <* USEDIN(SELF,
```

```
'PROCESS_PLANNING_SCHEMA.RELATIONSHIP_CONDITION.APPLICABLE_RELATIONSHIPS')
       | (rc.name = 'reason for revision') )) = 1;
  wr2: SIZEOF(QUERY ( ars <* USEDIN(SELF.related_action.chosen_method,
       'PROCESS_PLANNING_SCHEMA.ACTION_REQUEST_SOLUTION.METHOD') |
       (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST' IN
       TYPEOF(ars.request)) AND (ars.request.description =
       'revision level')) )) = 1;
  wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_APPROVAL_ASSIGNMENT.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.APPROVAL' IN TYPEOF(adr.
       assigned_approval)) )) >= 1;
  wr4: ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(SELF.
       related_action)) AND (
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_VERSION' IN TYPEOF(
       SELF.relating_action));
END_ENTITY; -- revision

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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
       = 0;
  wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
       SELF) |
    (('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'outer edge shape occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
```

 

```
      TYPEOF(sar))) ) | ((
      'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE' IN TYPEOF(sdr.
      relating_shape_aspect)) AND (sdr.relating_shape_aspect.
      description = 'outer edge shape')) )) = 1)) )) = 1)) )) = 0);
  wr6: (NOT (SELF\characterized_object.description = 'flat')) OR (
      SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'flat edge shape occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE'
      IN TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
      relating_shape_aspect.description = 'flat edge shape')) )) =
      1)) )) = 1)) )) = 0);
  wr7: (NOT (SELF\characterized_object.description = 'round')) OR (
      SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'rounded edge shape occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((
      'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
      TYPEOF(sdr.relating_shape_aspect)) AND (sdr.
      relating_shape_aspect.description = 'rounded edge shape')) ))
      = 1)) )) = 1)) )) = 0);
  wr8: (NOT (SELF\characterized_object.description = 'groove')) OR (
      SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'sweep occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'profile usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
      'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
      'PROCESS_PLANNING_SCHEMA.VEE_PROFILE',
      'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
      'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
      relating_shape_aspect)) = 1) AND (sdr.relating_shape_aspect.
      description = 'sweep')) )) = 1)) )) = 1)) )) = 0);
  wr9: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) |
     (('PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
```

```
       IN TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) <= 1;
END_ENTITY; -- revolved_profile

ENTITY rib_top
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'rib top condition occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'rib top usage') AND (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.RIB_TOP_FLOOR'
       IN TYPEOF(sdr.relating_shape_aspect)) AND (
       'PROCESS_PLANNING_SCHEMA.RIB_TOP' IN TYPEOF(sdr.
      related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
       = 0;
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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) |
      (('PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
       IN TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) <= 1;
END_ENTITY; -- rib_top

ENTITY rib_top_floor
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2: SELF.description IN ['planar','complex'];
  wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATING_SHAPE_ASPECT') | ((sar.description =
       'rib top usage') AND (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description =
       'rib top condition occurrence') AND (
       'PROCESS_PLANNING_SCHEMA.RIB_TOP' IN TYPEOF(fcr.
       related_shape_aspect.of_shape.definition))) AND (
       'PROCESS_PLANNING_SCHEMA.RIB_TOP_FLOOR' IN TYPEOF(fcr.
       relating_shape_aspect))) )) >= 1;
  wr4: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
       USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
```

 

```
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'boundary occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) ) | ((SIZEOF([
        'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
        'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
        'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
        'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE'] * TYPEOF(sdr.
        relating_shape_aspect)) = 1) AND (sdr.relating_shape_aspect.
        description = 'rib top floor boundary')) )) = 1)) )) = 1)) ))
        = 0);
END_ENTITY; -- rib_top_floor

ENTITY role_association;
  role      : object_role;
  item_with_role : role_select;
END_ENTITY; -- role_association

ENTITY round_hole
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'diameter occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) ) | ((
        'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
```

```
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'hole depth occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'bottom condition occurrence') AND (SIZEOF(
      QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'hole bottom usage') AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.HOLE_BOTTOM' IN
      TYPEOF(fcr.relating_shape_aspect)) AND (
      'PROCESS_PLANNING_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
     related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
      = 0;
  wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'change in diameter occurrence') AND (SIZEOF(
      QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description = 'taper usage')
      AND (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | (('PROCESS_PLANNING_SCHEMA.TAPER' IN
      TYPEOF(fcr.relating_shape_aspect)) AND (
      'PROCESS_PLANNING_SCHEMA.ROUND_HOLE' IN TYPEOF(fcr.
    related_shape_aspect.of_shape.definition))) )) = 1)) )) <= 1)) ))
      = 0;
  wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
      SELF) |
     ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) = 1;
END_ENTITY; -- round_hole


ENTITY rounded_end
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'partial circular boundary occurrence') AND (
        SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND (
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (
        'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE' IN
        TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
        description = 'course of travel occurrence') AND (SIZEOF(
        QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
        'RELATED_SHAPE_ASPECT') | ((sar.description =
        'path feature component usage') AND (
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
        TYPEOF(sar))) ) | ((
        'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(
        sdr.relating_shape_aspect)) AND (sdr.relating_shape_aspect.
        description = 'linear')) )) = 1)) )) = 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
        SELF) |
       (('PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION'
        IN TYPEOF(pdr.used_representation)) AND (pdr.
        used_representation.name = 'maximum feature limit')) )) <= 1;
END_ENTITY; -- rounded_end

ENTITY rounded_u_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
        TYPEOF(SELF.of_shape.definition);
  wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_SCHEMA.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
        used_representation)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
        (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
        'PROCESS_PLANNING_SCHEMA.' +
        'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
        'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (SIZEOF(QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.'
       + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
       = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'profile limit')) )) <= 1)) )) =
       0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'depth')) )) <= 1)) )) = 0)) ))
       = 0;
END_ENTITY; -- rounded_u_profile

ENTITY roundness_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1: NOT (('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- roundness_tolerance

ENTITY runout_zone_definition
 SUBTYPE OF (tolerance_zone_definition);
  orientation : runout_zone_orientation;
END_ENTITY; -- runout_zone_definition

ENTITY runout_zone_orientation;
```

 

```
  angle : measure_with_unit;
END_ENTITY; -- runout_zone_orientation

ENTITY runout_zone_orientation_reference_direction
 SUBTYPE OF (runout_zone_orientation);
  orientation_defining_relationship : shape_aspect_relationship;
END_ENTITY; -- runout_zone_orientation_reference_direction

ENTITY seam_curve
 SUBTYPE OF (surface_curve);
 WHERE
  wr1: SIZEOF(SELF\surface_curve.associated_geometry) = 2;
  wr2: associated_surface(SELF\surface_curve.associated_geometry[1]) =
       associated_surface(SELF\surface_curve.associated_geometry[2]);
  wr3: 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(SELF\surface_curve.
       associated_geometry[1]);
  wr4: 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(SELF\surface_curve.
       associated_geometry[2]);
END_ENTITY; -- seam_curve

ENTITY security_classification;
  name      : label;
  purpose    : text;
  security_level : security_classification_level;
END_ENTITY; -- security_classification

ENTITY security_classification_assignment
 ABSTRACT SUPERTYPE;
  assigned_security_classification : security_classification;
 DERIVE
  role : object_role := get_role(SELF);
 WHERE
  wr1: SIZEOF(USEDIN(SELF,'PROCESS_PLANNING_SCHEMA.' +
       'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
END_ENTITY; -- security_classification_assignment

ENTITY security_classification_level;
  name : label;
END_ENTITY; -- security_classification_level

ENTITY sequential_method
 SUBTYPE OF (serial_action_method);
  sequence_position : count_measure;
END_ENTITY; -- sequential_method

ENTITY serial_action_method
 SUBTYPE OF (action_method_relationship);
END_ENTITY; -- serial_action_method

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,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- shape_aspect
```

```
ENTITY shape_aspect_deriving_relationship
 SUBTYPE OF (shape_aspect_relationship);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.DERIVED_SHAPE_ASPECT' IN TYPEOF(SELF\
       shape_aspect_relationship.relating_shape_aspect);
END_ENTITY; -- shape_aspect_deriving_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,'PROCESS_PLANNING_SCHEMA.' +
       'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- shape_aspect_relationship

ENTITY shape_defining_relationship
 SUBTYPE OF (shape_aspect_relationship);
END_ENTITY; -- shape_defining_relationship

ENTITY shape_definition_representation
 SUBTYPE OF (property_definition_representation);
 WHERE
  wr1: ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN TYPEOF(
       SELF.definition)) OR (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINITION' IN TYPEOF(SELF.
       definition.definition));
  wr2: 'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION' IN TYPEOF(SELF.
       used_representation);
END_ENTITY; -- shape_definition_representation

ENTITY shape_dimension_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( temp <* SELF.items | (NOT (
       'PROCESS_PLANNING_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

ENTITY shape_representation
 SUBTYPE OF (representation);
END_ENTITY; -- shape_representation

ENTITY shape_representation_with_parameters
 SUBTYPE OF (shape_representation);
 WHERE
  wr1: SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.PLACEMENT',
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
       TYPEOF(it)) = 1)) )) = 0;
END_ENTITY; -- shape_representation_with_parameters
```

 

```
ENTITY shell_based_surface_model
 SUBTYPE OF (geometric_representation_item);
  sbsm_boundary : SET [1:?] OF shell;
 WHERE
  wr1: constraints_geometry_shell_based_surface_model(SELF);
END_ENTITY; -- shell_based_surface_model

ENTITY shell_based_wireframe_model
 SUBTYPE OF (geometric_representation_item);
  sbwm_boundary : SET [1:?] OF shell;
 WHERE
  wr1: constraints_geometry_shell_based_wireframe_model(SELF);
END_ENTITY; -- shell_based_wireframe_model

ENTITY shell_based_wireframe_shape_representation
 SUBTYPE OF (shape_representation);
 WHERE
  wr1 : SIZEOF(QUERY ( it <* SELF.items | (NOT (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM',
       'PROCESS_PLANNING_SCHEMA.AXIS2_PLACEMENT_3D'] * TYPEOF(it))
       = 1)) )) = 0;
  wr2 : SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL',
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM'] * TYPEOF(it)) = 1) ))
       >= 1;
  wr3 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
       sbwm\shell_based_wireframe_model.sbwm_boundary | (
       'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
       NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
       wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN
        TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
       edge_list | (NOT ('PROCESS_PLANNING_SCHEMA.EDGE_CURVE' IN
       TYPEOF(el.edge_element))) )) = 0)) )) = 0)) )) = 0)) )) = 0;
  wr4 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
       sbwm\shell_based_wireframe_model.sbwm_boundary | (
       'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
       NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
       wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN
        TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( pline_el <*
       QUERY ( el <* eloop\path.edge_list | (
       'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(el.
       edge_element\edge_curve.edge_geometry)) ) | (NOT (SIZEOF(
       pline_el.edge_element\edge_curve.edge_geometry\polyline.
       points) > 2)) )) = 0)) )) = 0)) )) = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
       TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
       sbwm\shell_based_wireframe_model.sbwm_boundary | (
       'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
       NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
       wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN
        TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
       edge_list | (NOT valid_wireframe_edge_curve(el.edge_element
       \edge_curve.edge_geometry)) )) = 0)) )) = 0)) )) = 0)) )) =
        0;
  wr6 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
```

```
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
      NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
      wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN
       TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
      edge_list | (NOT (('PROCESS_PLANNING_SCHEMA.VERTEX_POINT'
      IN TYPEOF(el.edge_element.edge_start)) AND (
      'PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(el.
      edge_element.edge_end)))) )) = 0)) )) = 0)) )) = 0)) )) = 0;
  wr7 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
      NOT (SIZEOF(QUERY ( eloop <* QUERY ( wsb <* ws\wire_shell.
      wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.EDGE_LOOP' IN
       TYPEOF(wsb)) ) | (NOT (SIZEOF(QUERY ( el <* eloop\path.
      edge_list | (NOT (valid_wireframe_vertex_point(el.
      edge_element.edge_start\vertex_point.vertex_geometry) AND
      valid_wireframe_vertex_point(el.edge_element.edge_end\
      vertex_point.vertex_geometry))) )) = 0)) )) = 0)) )) = 0)) ))
      = 0;
  wr8 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
      NOT (SIZEOF(QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
      wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.VERTEX_LOOP'
      IN TYPEOF(wsb)) ) | (NOT (
      'PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(vloop\
      vertex_loop.loop_vertex))) )) = 0)) )) = 0)) )) = 0;
  wr9 : SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( ws <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(sb)) ) | (
      NOT (SIZEOF(QUERY ( vloop <* QUERY ( wsb <* ws\wire_shell.
      wire_shell_extent | ('PROCESS_PLANNING_SCHEMA.VERTEX_LOOP'
      IN TYPEOF(wsb)) ) | (NOT valid_wireframe_vertex_point(vloop
      \vertex_loop.loop_vertex\vertex_point.vertex_geometry)) ))
      = 0)) )) = 0)) )) = 0;
  wr10: SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.VERTEX_SHELL' IN TYPEOF(sb)) ) | (
      NOT ('PROCESS_PLANNING_SCHEMA.VERTEX_POINT' IN TYPEOF(vs\
      vertex_shell.vertex_shell_extent.loop_vertex))) )) = 0)) ))
      = 0;
  wr11: SIZEOF(QUERY ( sbwm <* QUERY ( it <* SELF.items | (
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_MODEL' IN
      TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( vs <* QUERY ( sb <*
      sbwm\shell_based_wireframe_model.sbwm_boundary | (
      'PROCESS_PLANNING_SCHEMA.VERTEX_SHELL' IN TYPEOF(sb)) ) | (
      NOT valid_wireframe_vertex_point(vs\vertex_shell.
      vertex_shell_extent.loop_vertex\vertex_point.
      vertex_geometry)) )) = 0)) )) = 0;
  wr12: SIZEOF(QUERY ( mi <* QUERY ( it <* SELF.items | (
```

 

```
       'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM' IN TYPEOF(it)) ) | (
       NOT (('PROCESS_PLANNING_SCHEMA.' +
       'SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION') IN TYPEOF(mi\
       mapped_item.mapping_source.mapped_representation))) )) = 0;
  wr13: SELF.context_of_items\geometric_representation_context.
       coordinate_space_dimension = 3;
END_ENTITY; -- shell_based_wireframe_shape_representation

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

ENTITY single_activity_relationship
 SUBTYPE OF (sequential_method);
 WHERE
  wr2: 'PROCESS_PLANNING_SCHEMA.MANUFACTURING_PROCESS' IN TYPEOF(SELF.
       relating_method);
  wr3: SIZEOF(['PROCESS_PLANNING_SCHEMA.MACHINE_SETUP',
       'PROCESS_PLANNING_SCHEMA.TOOL_SETUP',
       'PROCESS_PLANNING_SCHEMA.FIXTURE_SETUP',
       'PROCESS_PLANNING_SCHEMA.ANCILLARY_SETUP'] * TYPEOF(SELF.
       related_method)) = 1;
END_ENTITY; -- single_activity_relationship

ENTITY slot
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'swept shape occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | (SIZEOF([
       'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
       'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
       'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
       'PROCESS_PLANNING_SCHEMA.VEE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
       'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE'] * TYPEOF(sdr.
       relating_shape_aspect)) = 1) )) = 1)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
```

```
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'course of travel occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND ((sar.name =
      'course of travel') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
      TYPEOF(sar)))) ) | (
      'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(
      sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'end condition occurrence') AND (SIZEOF(
      QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_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 (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((('PROCESS_PLANNING_SCHEMA.SLOT_END' IN
      TYPEOF(fcr.relating_shape_aspect)) AND (fcr.
      relating_shape_aspect.description IN ['open','radiused',
      'flat','woodruff'])) AND ('PROCESS_PLANNING_SCHEMA.SLOT' IN
      TYPEOF(fcr.related_shape_aspect.of_shape.definition))) )) =
      1)) )) = 2)) )) = 0;
wr5: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'end condition occurrence') AND (SIZEOF(
      QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_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 (
      'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
      TYPEOF(sar))) ) | ((('PROCESS_PLANNING_SCHEMA.SLOT_END' IN
      TYPEOF(fcr.relating_shape_aspect)) AND (fcr.
      relating_shape_aspect.description IN ['loop'])) AND (
      'PROCESS_PLANNING_SCHEMA.SLOT' IN TYPEOF(fcr.
     related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
      = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
      TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
      description = 'course of travel occurrence') AND (SIZEOF(
      QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
      'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
      'RELATED_SHAPE_ASPECT') | ((sar.description =
      'path feature component usage') AND ((sar.name =
      'course of travel') AND (
      'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
```

 

```
      TYPEOF(sar)))) ) | ((
      'PROCESS_PLANNING_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) | ((
      'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'maximum feature limit')) )) >= 0;
END_ENTITY; -- slot

ENTITY slot_end
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
  wr5 : (NOT (SELF.description IN ['flat'])) OR (SIZEOF(QUERY ( pd <*
      USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
      IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
      QUERY ( it <* impl_rep.used_representation.items | ((
      SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
       QUERY ( it <* impl_rep.used_representation.items | ((
       SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS'
       IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
       QUERY ( it <* impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_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 (
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
        TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
       = 'end condition occurrence') AND (
       'PROCESS_PLANNING_SCHEMA.SLOT' IN TYPEOF(fcr.
       related_shape_aspect.of_shape.definition))) AND (
       'PROCESS_PLANNING_SCHEMA.SLOT_END' IN TYPEOF(fcr.
       relating_shape_aspect))) )) >= 1;
END_ENTITY; -- slot_end

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

ENTITY solid_model
 SUPERTYPE OF (manifold_solid_brep)
```

 

```
 SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- solid_model

ENTITY spherical_cap
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
       = 0;
  wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'internal angle')) )) = 1)) ))
       = 0)) )) = 0;
END_ENTITY; -- spherical_cap

ENTITY spherical_surface
 SUBTYPE OF (elementary_surface);
  radius : positive_length_measure;
END_ENTITY; -- spherical_surface

ENTITY square_u_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
```

```
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
       = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) <= 1)) ))
```

 

```
     = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'second radius')) )) <= 1)) ))
     = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
     = 0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
     = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
     'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
     'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
     TYPEOF(pdr.used_representation)) AND (pdr.
     used_representation.name = 'profile limit')) )) <= 1)) )) =
      0;
wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'depth')) )) <= 1)) )) = 0)) ))
```

```
       = 0;

ENTITY standard_uncertainty
 SUBTYPE OF (uncertainty_qualifier);
  uncertainty_value : REAL;
END_ENTITY; -- standard_uncertainty

ENTITY step
 SUBTYPE OF (feature_definition);
 WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'course of travel occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'path feature component usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ((
       'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT' IN TYPEOF(
       sdr.relating_shape_aspect)) AND (sdr.relating_shape_aspect.
       description = 'linear')) )) = 1)) )) = 1)) )) = 0;
  wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ.
       description = 'removal boundary occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'profile usage') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.VEE_PROFILE' IN
       TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
       SELF) | ((
       'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'maximum feature limit')) )) >= 0;
  wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (((
       'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT' IN TYPEOF(
       csa)) AND (csa.name = 'uncut area')) AND (SIZEOF(
       QUERY ( sar <* csa.component_relationships | ((
       'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP' IN
```

 

```
       TYPEOF(sar)) AND (SIZEOF(['PROCESS_PLANNING_SCHEMA.BOSS',
       'PROCESS_PLANNING_SCHEMA.PROTRUSION'] * TYPEOF(sar.
       related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
END_ENTITY; -- step

ENTITY straightness_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1: NOT (('PROCESS_PLANNING_SCHEMA.' +
       'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
END_ENTITY; -- straightness_tolerance

ENTITY surface
 SUPERTYPE OF (ONEOF (elementary_surface,swept_surface,bounded_surface,
   offset_surface,surface_replica))
 SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- surface

ENTITY surface_curve
 SUPERTYPE OF (ONEOF (intersection_curve,seam_curve) ANDOR
   bounded_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: ('PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(associated_geometry
       [1])) OR (master_representation <> pcurve_s1);
  wr3: ('PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(associated_geometry
       [2])) OR (master_representation <> pcurve_s2);
  wr4: NOT ('PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(curve_3d));
END_ENTITY; -- surface_curve

ENTITY surface_of_linear_extrusion
 SUBTYPE OF (swept_surface);
  extrusion_axis : vector;
END_ENTITY; -- surface_of_linear_extrusion

ENTITY surface_of_revolution
 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

ENTITY surface_patch
 SUBTYPE OF (founded_item);
  parent_surface : bounded_surface;
  u_transition  : transition_code;
  v_transition  : transition_code;
  u_sense    : BOOLEAN;
  v_sense    : BOOLEAN;
 INVERSE
  using_surfaces : BAG [1:?] OF rectangular_composite_surface FOR
           segments;
 WHERE
  wr1: NOT ('PROCESS_PLANNING_SCHEMA.CURVE_BOUNDED_SURFACE' IN TYPEOF(
```

```
       parent_surface));
END_ENTITY; -- surface_patch

ENTITY surface_profile_tolerance
 SUBTYPE OF (geometric_tolerance);
 WHERE
  wr1: (NOT (('PROCESS_PLANNING_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

ENTITY surface_replica
 SUBTYPE OF (surface);
  parent_surface : surface;
  transformation : cartesian_transformation_operator_3d;
 WHERE
  wr1: acyclic_surface_replica(SELF,parent_surface);
END_ENTITY; -- surface_replica

ENTITY swept_surface
 SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
 SUBTYPE OF (surface);
  swept_curve : curve;
END_ENTITY; -- swept_surface

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 | (
       'PROCESS_PLANNING_SCHEMA.CENTRE_OF_SYMMETRY' IN TYPEOF(x.
       related_shape_aspect)) )) >= 1;
END_ENTITY; -- symmetric_shape_aspect

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

ENTITY tangent
 SUBTYPE OF (derived_shape_aspect);
 WHERE
  wr1: SIZEOF(SELF\derived_shape_aspect.deriving_relationships) = 1;
END_ENTITY; -- tangent

ENTITY taper
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: 'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
```

 

```
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) )) = 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
       'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION' IN
       TYPEOF(pdr.used_representation)) AND (pdr.
       used_representation.name = 'direction')) )) = 1)) )) = 0);
END_ENTITY; -- taper

ENTITY tee_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
```

 

```
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
      = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'depth')) )) = 1)) )) = 0)) ))
      = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'cross bar width')) )) = 1)) ))
      = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'cross bar depth')) )) = 1)) ))
      = 0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
      = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

```
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) = 1)) ))
      = 0)) )) = 0;
  wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
      = 0)) )) = 0;
  wr13: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
      = 0)) )) = 0;
  wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
     TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) = 0)) ))
      = 0;
  wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
      'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'profile limit')) )) <= 1)) )) =
       0;
END_ENTITY; -- tee_profile
```

 

```
ENTITY thread
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
       = 0)) )) = 0;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'minor diameter')) )) <= 1)) ))
       = 0)) )) = 0;
  wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'pitch diameter')) )) <= 1)) ))
       = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
```

```
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.RATIO_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
      1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
      = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) ))
      = 0;
  wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'hand')) AND (it.description
      IN ['left','right'])) )) = 1)) )) = 0)) )) = 0;
  wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) =
      0)) )) = 0;
  wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
```

 

```
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_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,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((SIZEOF([
     'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
     'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
    TYPEOF(it)) = 2) AND (it.name = 'crest')) )) <= 1)) )) = 0)) ))
     = 0;
wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
     .description = 'partial area occurrence') AND (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATED_SHAPE_ASPECT') | ((sar.description =
     'applied area usage') AND (
     'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
     TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.APPLIED_AREA'
     IN TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) ))
     = 0;
wr13: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
     TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
     QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
     'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
     'RELATED_SHAPE_ASPECT') | ((sar.description =
     'applied shape') AND (
     'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
     TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
     IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
      0;
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
     'PROCESS_PLANNING_SCHEMA.' +
     'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
     used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
     impl_rep.used_representation.items | ((
     'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
     IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
     = 0)) )) = 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
      | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
     pd,'PROCESS_PLANNING_SCHEMA.' +
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'nominal size')) )) <= 1)) ))
       = 0)) )) = 0;
END_ENTITY; -- thread

ENTITY time_measure_with_unit
 SUBTYPE OF (measure_with_unit);
 WHERE
  wr1: 'PROCESS_PLANNING_SCHEMA.TIME_UNIT' IN TYPEOF(SELF\
       measure_with_unit.unit_component);
END_ENTITY; -- time_measure_with_unit

ENTITY time_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 = 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

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

ENTITY tolerance_zone
 SUBTYPE OF (shape_aspect);
  defining_tolerance : SET [1:?] OF geometric_tolerance;
  form        : tolerance_zone_form;
END_ENTITY; -- tolerance_zone

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

ENTITY tolerance_zone_form;
  name : label;
END_ENTITY; -- tolerance_zone_form

ENTITY tool_assembly
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.TOOL_CONTRACT' IN TYPEOF(adr.
       assigned_contract)) )) <= 1;
```

 

```
  wr2: SIZEOF(QUERY ( ar <* QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
       | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
       TYPEOF(adr)) ) | (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       'PROCESS_PLANNING_SCHEMA.TOOL_PLACEMENT_INSTRUCTION' IN
       TYPEOF(rp)) )) = 1) )) >= 0;
  wr3: SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | (NOT (SIZEOF(QUERY ( it <* impl_rep.
       representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
       TYPEOF(it)) = 1) AND (it.name =
       'tool assembly tool parameters')) )) >= 0)) )) = 0)) )) = 0;
  wr4: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.RELATING_RESOURCE')
       | ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP' IN
       TYPEOF(arr)) ) | (
       'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT' IN TYPEOF(
       ar2.related_resource)) )) >= 1;
  wr5: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(adr
       .assigned_document)) )) <= 1;
  wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
       | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
       TYPEOF(ar)) ) | (NOT (SIZEOF(QUERY ( pdp_imp <*
       QUERY ( pdp <* arr.operations | (
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS' IN
       TYPEOF(pdp)) ) | (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
    'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
       property_or_shape)) AND (ppa.description =
       'tool assembly shape')) )) >= 0) )) = 0)) )) = 0;
  wr7: NOT (SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | ((impl_rep.representation.name =
       'tool assembly parameter') AND (SIZEOF(QUERY ( it <*
       impl_rep.representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_SCHEMA.COUNT_MEASURE' IN
       TYPEOF(it\measure_with_unit.value_component))) AND (it.name
       = 'number of spares')) )) <= 1)) )) = 0)) )) = 0);
END_ENTITY; -- tool_assembly

ENTITY tool_assembly_element
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | (adr.assigned_document.description =
       'tool catalogue number') )) <= 1;
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
```

```
    'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
       | (adr.assigned_organization.description =
       'tool company name') )) = 1;
  wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (adr
       .description = 'tool element identification') )) = 1;
  wr4: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
       TYPEOF(adr.assigned_document)) )) >= 0;
  wr5: NOT (SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       SIZEOF(QUERY ( apr <* USEDIN(ap,
  'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | (('PROCESS_PLANNING_SCHEMA.' +
       'EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS') IN
       TYPEOF(apr)) )) <= 1) )) = 0);
  wr6: SIZEOF(QUERY ( arr <* QUERY ( ar <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE.RESOURCES')
       | ('PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT' IN
       TYPEOF(ar)) ) | (NOT (SIZEOF(QUERY ( pdp_imp <*
       QUERY ( pdp <* arr.operations | (
       'PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_PROCESS' IN
       TYPEOF(pdp)) ) | (SIZEOF(QUERY ( ppa <* USEDIN(pdp_imp,
     'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
       | (('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT' IN TYPEOF(ppa.
       property_or_shape)) AND (ppa.description =
       'tool assembly element shape')) )) = 0) )) = 0)) )) >= 0;
END_ENTITY; -- tool_assembly_element

ENTITY tool_assembly_relationship
 SUBTYPE OF (action_resource_relationship);
 WHERE
  wr1: (('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(SELF.
       relating_resource)) AND (
       'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT' IN TYPEOF(
       SELF.related_resource))) OR ((
       'PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL' IN
       TYPEOF(SELF.relating_resource)) AND (
       'PROCESS_PLANNING_SCHEMA.WORKSTATION' IN TYPEOF(SELF.
       related_resource)));
END_ENTITY; -- tool_assembly_relationship

ENTITY tool_body
 SUBTYPE OF (tool_assembly_element);
 WHERE
  wr1: SIZEOF(QUERY ( rp <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
       NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(rp,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | ('PROCESS_PLANNING_SCHEMA.REPRESENTATION' IN TYPEOF(pdr.
       representation)) ) | (NOT (SIZEOF(QUERY ( it <* impl_rep.
       representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'] *
       TYPEOF(it)) = 1) AND (it.name = 'tool body parameter')) ))
       >= 0)) )) = 0)) )) = 0;
END_ENTITY; -- tool_body

ENTITY tool_contract
 SUBTYPE OF (contract);
```

 

```
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST' IN
       TYPEOF(adr.assigned_action_request)) AND (adr.
       assigned_action_request.description = 'design order')) )) =
       1;
  wr2: SIZEOF(QUERY ( adr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.APPLIED_ACTION_REQUEST_ASSIGNMENT.ITEMS')
       | (('PROCESS_PLANNING_SCHEMA.VERSIONED_ACTION_REQUEST' IN
       TYPEOF(adr.assigned_action_request)) AND (adr.
       assigned_action_request.description = 'fabrication order')) ))
       = 1;
  wr3: SIZEOF(QUERY ( ap <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.CONTRACT_ASSIGNMENT.ASSIGNED_CONTRACT')
       | (('PROCESS_PLANNING_SCHEMA.APPLIED_CONTRACT_ASSIGNMENT'
       IN TYPEOF(ap)) AND (NOT (SIZEOF(QUERY ( it <* ap.items | ((
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
       TYPEOF(it)) AND (it\document.description =
       'tool contract document')) )) <= 1))) )) = 0;
END_ENTITY; -- tool_contract

ENTITY tool_magazine_turret_carousel
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (adr
       .description = 'tool capacity') )) = 1;
  wr2: SIZEOF(QUERY ( ar2 <* QUERY ( arr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_RELATIONSHIP.' +
       'RELATING_RESOURCE') | (
       'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP' IN
       TYPEOF(arr)) ) | ('PROCESS_PLANNING_SCHEMA.WORKSTATION' IN
       TYPEOF(ar2.related_resource)) )) = 1;
END_ENTITY; -- tool_magazine_turret_carousel

ENTITY tool_placement_instruction
 SUBTYPE OF (resource_property);
 WHERE
  wr1: SIZEOF(QUERY ( rpr <* USEDIN(SELF,
   'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY_REPRESENTATION.PROPERTY')
       | ((rpr.name = 'tool position') AND (SIZEOF(QUERY ( it <*
       rpr.representation.items | ((
       'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (it.name = 'tool position')) )) = 1)) )) =
       0;
  wr2: 'PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL' IN
       TYPEOF(SELF.resource);
  wr3: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | ('PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION' IN
       TYPEOF(adr.assigned_document)) )) = 1;
END_ENTITY; -- tool_placement_instruction

ENTITY tool_setup
 SUBTYPE OF (process_plan_activity);
 WHERE
  wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
```

```
       TYPEOF(arr)) ) | (NOT (SIZEOF(QUERY ( it <* am.resources |
       ('PROCESS_PLANNING_SCHEMA.MACHINE' IN TYPEOF(it)) )) = 1)) ))
       = 0;
  wr2: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (NOT (SIZEOF(QUERY ( it <* am.resources |
       ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)) )) =
       1)) )) = 0;
  wr3: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (NOT (SIZEOF(QUERY ( it <* am.resources |
       ('PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)) )) =
       1)) )) = 0;
END_ENTITY; -- tool_setup

ENTITY topological_representation_item
 SUPERTYPE OF (ONEOF (vertex,edge,face_bound,face,vertex_shell,
   wire_shell,connected_edge_set,connected_face_set,loop ANDOR path))
 SUBTYPE OF (representation_item);
END_ENTITY; -- topological_representation_item

ENTITY toroidal_surface
 SUBTYPE OF (elementary_surface);
  major_radius : positive_length_measure;
  minor_radius : positive_length_measure;
END_ENTITY; -- toroidal_surface

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

ENTITY transition_feature
 SUPERTYPE OF (ONEOF (chamfer,edge_round,fillet))
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1: SIZEOF(['PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION',
       'PROCESS_PLANNING_SCHEMA.COMPOUND_FEATURE'] * TYPEOF(SELF.
       of_shape.definition)) = 1;
  wr2: SIZEOF(['PROCESS_PLANNING_SCHEMA.CHAMFER',
       'PROCESS_PLANNING_SCHEMA.EDGE_ROUND',
       'PROCESS_PLANNING_SCHEMA.FILLET'] * TYPEOF(SELF)) = 1;
END_ENTITY; -- transition_feature

ENTITY trimmed_curve
 SUBTYPE OF (bounded_curve);
  basis_curve      : curve;
  trim_1        : SET [1:2] OF trimming_select;
  trim_2        : SET [1:2] OF trimming_select;
  sense_agreement    : BOOLEAN;
  master_representation : trimming_preference;
 WHERE
  wr1: (HIINDEX(trim_1) = 1) OR (TYPEOF(trim_1[1]) <> TYPEOF(trim_1[2]));
  wr2: (HIINDEX(trim_2) = 1) OR (TYPEOF(trim_2[1]) <> TYPEOF(trim_2[2]));
END_ENTITY; -- trimmed_curve
```

 

```
ENTITY turned_knurl
 SUBTYPE OF (feature_definition);
 WHERE
  wr1 : SELF\characterized_object.description IN ['diamond','diagonal',
       'straight'];
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND ('PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
       = 0)) )) = 0;
  wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((SIZEOF([
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
       'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'nominal diameter')) )) = 1)) ))
       = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.' +
       'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
       QUERY ( pdr <* USEDIN(pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'tooth depth')) )) <= 1)) ))
      = 0)) )) = 0;
  wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'root fillet')) )) <= 1)) ))
      = 0)) )) = 0;
  wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((
      'PROCESS_PLANNING_SCHEMA.DESCRIPTIVE_REPRESENTATION_ITEM'
      IN TYPEOF(it)) AND (it.name = 'helix hand')) )) = 1)) )) =
      0)) )) = 0);
  wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
```

 

```
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | ((sa_occ
       .description = 'partial area occurrence') AND (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'applied area usage') AND (('PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
       'PROCESS_PLANNING_SCHEMA.APPLIED_AREA' IN TYPEOF(sdr.
       relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
  wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | ('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
       TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT.OF_SHAPE') | (SIZEOF(
       QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
       'PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT_RELATIONSHIP.' +
       'RELATED_SHAPE_ASPECT') | ((sar.description =
       'applied shape') AND (
       'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP' IN
       TYPEOF(sar))) ) | ('PROCESS_PLANNING_SCHEMA.SHAPE_ASPECT'
       IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
        0;
END_ENTITY; -- turned_knurl

ENTITY type_qualifier;
  name : label;
END_ENTITY; -- type_qualifier

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

ENTITY uncertainty_qualifier
 SUPERTYPE OF (standard_uncertainty);
  measure_name : label;
  description : text;
END_ENTITY; -- uncertainty_qualifier

ENTITY uniform_curve
 SUBTYPE OF (b_spline_curve);
END_ENTITY; -- uniform_curve

ENTITY uniform_surface
 SUBTYPE OF (b_spline_surface);
END_ENTITY; -- uniform_surface

ENTITY validation
 SUBTYPE OF (manufacturing_activity);
 WHERE
  wr1: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
    'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
       | (
       'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
       TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
       'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY' IN TYPEOF(it)) ))
```

```
        <= 1) )) = 1;
   wr2: SIZEOF(QUERY ( am <* QUERY ( arr <* USEDIN(SELF,
     'PROCESS_PLANNING_SCHEMA.ACTION_RESOURCE_REQUIREMENT.OPERATIONS')
        | (
        'PROCESS_PLANNING_SCHEMA.REQUIREMENT_FOR_ACTION_RESOURCE' IN
        TYPEOF(arr)) ) | (SIZEOF(QUERY ( it <* am.resources | (
        'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY' IN TYPEOF(it)) )) =
        1) )) = 1;
   wr3: SIZEOF(QUERY ( cm <* USEDIN(SELF,
        'PROCESS_PLANNING_SCHEMA.ACTION.CHOSEN_METHOD') | (NOT (
        SIZEOF(QUERY ( ppa <* USEDIN(cm,
      'PROCESS_PLANNING_SCHEMA.PROCESS_PROPERTY_ASSOCIATION.PROCESS')
        | (('PROCESS_PLANNING_SCHEMA.PRODUCT_DEFINITION_SHAPE' IN
        TYPEOF(ppa.property_or_shape)) AND (ppa.description =
        'part fixture mounting part shape')) )) = 1)) )) = 1;
   wr4: SIZEOF(QUERY ( adr <* SELF\
        action_method_with_associated_documents.documents | (
       'PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM' IN TYPEOF(adr)) ))
        <= 1;
 END_ENTITY; -- validation

 ENTITY value_range
  SUBTYPE OF (compound_representation_item);
 WHERE

wr1: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
   ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri))
    ))=2;

wr2: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
 (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri)) AND
   (mri.name='lower limit'))
    ))=1;

wr3: SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
   ('PROCESS_PLANNING_SCHEMA.SET_REPRESENTATION_ITEM' IN TYPEOF (sri))) |
 (('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (mri)) AND
   (mri.name='upper limit'))
    ))=1;

wr4: SIZEOF(QUERY( i1 <* SELF.item_element |
 ('PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN TYPEOF (i1)) AND
   (SIZEOF ( QUERY (i2 <* SELF.item_element |
 ('PROCESS_PLANNING_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

 ENTITY value_representation_item
  SUBTYPE OF (representation_item);
   value_component : measure_value;
  WHERE
   wr1: SIZEOF(QUERY ( rep <* using_representations(SELF) | (NOT (
        'PROCESS_PLANNING_SCHEMA.GLOBAL_UNIT_ASSIGNED_CONTEXT' IN
        TYPEOF(rep.context_of_items))) )) = 0;
 END_ENTITY; -- value_representation_item
```

 

```
ENTITY vector
 SUBTYPE OF (geometric_representation_item);
  orientation : direction;
  magnitude  : length_measure;
 WHERE
  wr1: magnitude >= 0;
END_ENTITY; -- vector

ENTITY vee_profile
 SUBTYPE OF (shape_aspect);
 WHERE
  wr1 : 'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION' IN
       TYPEOF(SELF.of_shape.definition);
  wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) )) = 1)) )) = 0;
  wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
       'PROCESS_PLANNING_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,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | ((
       'PROCESS_PLANNING_SCHEMA.PLACEMENT' IN TYPEOF(it)) AND (it.
       name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
        | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'profile angle')) )) = 1)) ))
      = 0)) )) = 0;
  wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | ((SIZEOF([
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM',
      'PROCESS_PLANNING_SCHEMA.PLANE_ANGLE_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'tilt angle')) )) = 1)) ))
      = 0)) )) = 0;
  wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION' IN
      TYPEOF(pdr.used_representation)) AND (pdr.
      used_representation.name = 'profile limit')) )) <= 1)) )) =
       0;
  wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
      pd,'PROCESS_PLANNING_SCHEMA.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
      'PROCESS_PLANNING_SCHEMA.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
      used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
      impl_rep.used_representation.items | (((
      'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
      TYPEOF(it)) AND (
      'PROCESS_PLANNING_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,
      'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION')
       | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```

 

```
       pd,'PROCESS_PLANNING_SCHEMA.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
       'PROCESS_PLANNING_SCHEMA.' +
       'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
       used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
       impl_rep.used_representation.items | (((
       'PROCESS_PLANNING_SCHEMA.MEASURE_REPRESENTATION_ITEM' IN
       TYPEOF(it)) AND (
       'PROCESS_PLANNING_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

ENTITY versioned_action_request;
  id     : identifier;
  version   : label;
  purpose   : text;
  description : OPTIONAL text;
END_ENTITY; -- versioned_action_request

ENTITY vertex
 SUBTYPE OF (topological_representation_item);
END_ENTITY; -- vertex

ENTITY vertex_loop
 SUBTYPE OF (loop);
  loop_vertex : vertex;
END_ENTITY; -- vertex_loop

ENTITY vertex_point
 SUBTYPE OF (vertex, geometric_representation_item);
  vertex_geometry : point;
END_ENTITY; -- vertex_point

ENTITY vertex_shell
 SUBTYPE OF (topological_representation_item);
  vertex_shell_extent : vertex_loop;
END_ENTITY; -- vertex_shell

ENTITY view_reference
 SUBTYPE OF (document_usage_constraint);
 WHERE
  wr1: NOT (SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF.source,
     'PROCESS_PLANNING_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
       | ('PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE' IN
       TYPEOF(dr)) ) | (SIZEOF(QUERY ( d <* adr.items | (
      'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(d)) ))
       >= 0) )) = 0);
  wr2: 'PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE' IN TYPEOF(SELF.
       source);
  wr3: SELF.subject_element IN ['sheet','view','zone'];
END_ENTITY; -- view_reference

ENTITY wire_shell
 SUBTYPE OF (topological_representation_item);
  wire_shell_extent : SET [1:?] OF loop;
 WHERE
  wr1: NOT mixed_loop_type_set(wire_shell_extent);
END_ENTITY; -- wire_shell

ENTITY work_cell
```

```
 SUBTYPE OF (applied_group_assignment, action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* SELF.items | (
       'PROCESS_PLANNING_SCHEMA.WORKSTATION' IN TYPEOF(adr)) )) >=
       1;
  wr2: SIZEOF(QUERY ( adr <* SELF.usage | (
       'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_ACTIVITY' IN TYPEOF(
       adr)) )) >= 0;
  wr3: SIZEOF(QUERY ( ar <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.RESOURCE_PROPERTY.RESOURCE') | (
      'PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION' IN TYPEOF(ar)) ))
       = 1;
END_ENTITY; -- work_cell

ENTITY workstation
 SUBTYPE OF (action_resource);
 WHERE
  wr1: SIZEOF(QUERY ( adr <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_DOCUMENT_REFERENCE.ITEMS')
       | (adr.assigned_document.kind.product_data_type =
       'special capability') )) >= 0;
  wr2: SIZEOF(QUERY ( wc <* USEDIN(SELF,
       'PROCESS_PLANNING_SCHEMA.APPLIED_GROUP_ASSIGNMENT.ITEMS') |
       ('PROCESS_PLANNING_SCHEMA.WORK_CELL' IN TYPEOF(wc)) )) = 1;
END_ENTITY; -- workstation

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 =
      'process_planning_schema')) )) = 1)) )) = 0;

END_RULE; -- application_context_requires_ap_definition

RULE approval_requires_approval_date_time FOR (approval,
      approval_date_time);

WHERE
 wr1: SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( adt <*
      approval_date_time | (app :=: adt.dated_approval) )) = 1)) ))
      = 0;

END_RULE; -- approval_requires_approval_date_time

RULE approval_requires_approval_person_organization FOR (approval,
      approval_person_organization);

WHERE
 wr1: SIZEOF(QUERY ( app <* approval | (NOT (SIZEOF(QUERY ( apo <*
      approval_person_organization | (app :=: apo.
      authorized_approval) )) >= 1)) )) = 0;

END_RULE; -- approval_requires_approval_person_organization

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

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

RULE dependent_instantiable_date FOR (date);

WHERE
 wr1: SIZEOF(QUERY ( dt <* date | (NOT (SIZEOF(USEDIN(dt,'')) >= 1)) ))
      = 0;

END_RULE; -- dependent_instantiable_date

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

RULE dependent_instantiable_precision_qualifier FOR (precision_qualifier);

WHERE
 wr1: SIZEOF(QUERY ( pq <* precision_qualifier | (NOT (SIZEOF(USEDIN(pq,
      '')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_precision_qualifier

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

 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
```

```
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

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

RULE externally_defined_class_with_known_source_requirement FOR (
          externally_defined_class);

WHERE
  wr1: SIZEOF(QUERY ( edc <* externally_defined_class | ((
          'PROCESS_PLANNING_SCHEMA.KNOWN_SOURCE' IN TYPEOF(edc.source))
          AND (NOT (SIZEOF(QUERY ( aoa <* USEDIN(edc.source,
        'PROCESS_PLANNING_SCHEMA.APPLIED_ORGANIZATION_ASSIGNMENT.ITEMS')
           | (aoa.role.name = 'library supplier') )) = 1))) )) <= 1;

END_RULE; -- externally_defined_class_with_known_source_requirement

RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);

WHERE
  wr1: SIZEOF(QUERY ( gt <* geometric_tolerance | (NOT (SIZEOF(TYPEOF(gt)
          * ['PROCESS_PLANNING_SCHEMA.ANGULARITY_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.CIRCULAR_RUNOUT_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.CONCENTRICITY_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.CYLINDRICITY_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.FLATNESS_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.LINE_PROFILE_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.PARALLELISM_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.PERPENDICULARITY_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.POSITION_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.ROUNDNESS_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.STRAIGHTNESS_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.SURFACE_PROFILE_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.SYMMETRY_TOLERANCE',
          'PROCESS_PLANNING_SCHEMA.TOTAL_RUNOUT_TOLERANCE']) <= 2)) )) =
           0;

END_RULE; -- geometric_tolerance_subtype_exclusiveness

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

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,'PROCESS_PLANNING_SCHEMA.' +
          'MATERIAL_DESIGNATION.DEFINITIONS')) = 0) ) | (NOT (SIZEOF(
          QUERY ( mfuo <* make_from_usage_option | (NOT (nmpd :=: mfuo.
          relating_product_definition)) )) >= 1)) )) = 0;

END_RULE; -- material_is_specified_for_part

RULE mating_definition_relationship_orientation FOR (
          product_definition_relationship);

WHERE
  wr1: SIZEOF(QUERY ( prd_def_rel <* product_definition_relationship | (
          NOT (SIZEOF(QUERY ( pd <* USEDIN(prd_def_rel,
          'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION.DEFINITION') | (
          NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
 'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')
           | (((
          'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINITION_REPRESENTATION' IN
          TYPEOF(pdr)) AND (
          'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION' IN TYPEOF(pdr.
          used_representation))) AND (NOT (SIZEOF(QUERY ( rm <* USEDIN(
          pdr.used_representation,
      'PROCESS_PLANNING_SCHEMA.REPRESENTATION_MAP.MAPPED_REPRESENTATION')
           | (SIZEOF(QUERY ( mi <* USEDIN(rm,
          'PROCESS_PLANNING_SCHEMA.MAPPED_ITEM.MAPPING_SOURCE') | (
          'PROCESS_PLANNING_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(mi.
          mapping_target)) )) <= 1) )) = 0))) )) = 0)) )) = 0)) )) = 0;

END_RULE; -- mating_definition_relationship_orientation

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

RULE shape_aspect_relationship_subtype_exclusiveness FOR (
          shape_aspect_relationship);

WHERE
  wr1: SIZEOF(QUERY ( sr <* shape_aspect_relationship | (NOT (SIZEOF(
          TYPEOF(sr) * ['PROCESS_PLANNING_SCHEMA.DIMENSIONAL_LOCATION',
          'PROCESS_PLANNING_SCHEMA.GEOMETRIC_TOLERANCE_RELATIONSHIP',
          'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_RELATIONSHIP',
        'PROCESS_PLANNING_SCHEMA.SHAPE_DEFINING_RELATIONSHIP']) <= 2)) ))
          = 0;

END_RULE; -- shape_aspect_relationship_subtype_exclusiveness

RULE shape_representation_subtype_exclusiveness FOR (
          shape_representation);

WHERE
wr1: SIZEOF(QUERY ( sr <* shape_representation | (NOT (SIZEOF(TYPEOF(sr) *
```

```
            ['PROCESS_PLANNING_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
           'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
           'PROCESS_PLANNING_SCHEMA.SHAPE_DIMENSION_REPRESENTATION'])
            <= 2)) )) = 0;

END_RULE; -- shape_representation_subtype_exclusiveness

RULE subtype_exclusiveness_action_resource FOR (action_resource);

WHERE
  wr1: SIZEOF(QUERY ( a <* action_resource | (NOT type_check_function(a,
          ['PROCESS_PLANNING_SCHEMA.CONTROLLER',
          'PROCESS_PLANNING_SCHEMA.GENERIC_MANUFACTURING_RESOURCE',
          'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY',
          'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_ELEMENT',
          'PROCESS_PLANNING_SCHEMA.WORK_CELL',
          'PROCESS_PLANNING_SCHEMA.MACHINE',
          'PROCESS_PLANNING_SCHEMA.PALLET',
          'PROCESS_PLANNING_SCHEMA.WORKSTATION',
          'PROCESS_PLANNING_SCHEMA.TOOL_MAGAZINE_TURRET_CAROUSEL',
          'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY',
          'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_ELEMENT'],3)) )) = 0;

END_RULE; -- subtype_exclusiveness_action_resource

RULE subtype_exclusiveness_action_resource_relationship FOR (
          action_resource_relationship);

WHERE
  wr1: SIZEOF(QUERY ( a <* action_resource_relationship | (NOT
          type_check_function(a,[
          'PROCESS_PLANNING_SCHEMA.MACHINE_ELEMENT_RELATIONSHIP',
          'PROCESS_PLANNING_SCHEMA.FIXTURE_ASSEMBLY_RELATIONSHIP',
          'PROCESS_PLANNING_SCHEMA.TOOL_ASSEMBLY_RELATIONSHIP'],3)) )) =
           0;

END_RULE; -- subtype_exclusiveness_action_resource_relationship

RULE subtype_exclusiveness_characterized_object FOR (
          characterized_object);

WHERE
  wr1: SIZEOF(QUERY ( a <* characterized_object | (NOT
          type_check_function(a,['PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE',
          'PROCESS_PLANNING_SCHEMA.FEATURE_DEFINITION',
           'PROCESS_PLANNING_SCHEMA.FEATURE_COMPONENT_DEFINITION'],3)) ))
          = 0;

END_RULE; -- subtype_exclusiveness_characterized_object

RULE subtype_exclusiveness_contract FOR (contract);

WHERE
  wr1: SIZEOF(QUERY ( a <* contract | (NOT type_check_function(a,[
          'PROCESS_PLANNING_SCHEMA.FIXTURE_CONTRACT',
          'PROCESS_PLANNING_SCHEMA.PART_CONTRACT',
          'PROCESS_PLANNING_SCHEMA.TOOL_CONTRACT'],3)) )) = 0;

END_RULE; -- subtype_exclusiveness_contract

RULE subtype_exclusiveness_document FOR (document);
```

 

```
WHERE
  wr1: SIZEOF(QUERY ( a <* document | (NOT type_check_function(a,[
           'PROCESS_PLANNING_SCHEMA.PROCESS_PLAN_SPECIFICATION',
           'PROCESS_PLANNING_SCHEMA.DOCUMENT_FILE',
           'PROCESS_PLANNING_SCHEMA.DESIGN_REFERENCE',
           'PROCESS_PLANNING_SCHEMA.CONTROLLER_PROGRAM',
           'PROCESS_PLANNING_SCHEMA.ILLUSTRATION'],4)) )) = 0;

END_RULE; -- subtype_exclusiveness_document

RULE subtype_exclusiveness_group FOR (group);

WHERE
  wr1: SIZEOF(QUERY ( a <* group | (NOT type_check_function(a,[
           'PROCESS_PLANNING_SCHEMA.CLASS'],3)) )) = 0;

END_RULE; -- subtype_exclusiveness_group

RULE subtype_exclusiveness_resource_property FOR (resource_property);

WHERE
  wr1: SIZEOF(QUERY ( a <* resource_property | (NOT type_check_function(
           a,['PROCESS_PLANNING_SCHEMA.PART_HOLDING_POSITION',
           'PROCESS_PLANNING_SCHEMA.TOOL_PLACEMENT_INSTRUCTION',
           'PROCESS_PLANNING_SCHEMA.IN_FACILITY_LOCATION'],3)) )) = 0;

END_RULE; -- subtype_exclusiveness_resource_property

RULE subtype_exclusiveness_shape_aspect FOR (shape_aspect);

WHERE
  wr1: SIZEOF(QUERY ( a <* shape_aspect | (NOT type_check_function(a,[
           'PROCESS_PLANNING_SCHEMA.APPLIED_AREA',
           'PROCESS_PLANNING_SCHEMA.BOSS_TOP',
           'PROCESS_PLANNING_SCHEMA.CHAMFER_OFFSET',
           'PROCESS_PLANNING_SCHEMA.CIRCULAR_CLOSED_PROFILE',
           'PROCESS_PLANNING_SCHEMA.CLOSED_PATH_PROFILE',
           'PROCESS_PLANNING_SCHEMA.COMPOSITE_SHAPE_ASPECT',
           'PROCESS_PLANNING_SCHEMA.DATUM',
           'PROCESS_PLANNING_SCHEMA.DATUM_FEATURE',
           'PROCESS_PLANNING_SCHEMA.DATUM_TARGET',
           'PROCESS_PLANNING_SCHEMA.DERIVED_SHAPE_ASPECT',
           'PROCESS_PLANNING_SCHEMA.HOLE_BOTTOM',
           'PROCESS_PLANNING_SCHEMA.LINEAR_PROFILE',
           'PROCESS_PLANNING_SCHEMA.MODIFIED_PATTERN',
           'PROCESS_PLANNING_SCHEMA.NGON_CLOSED_PROFILE',
           'PROCESS_PLANNING_SCHEMA.OPEN_PATH_PROFILE',
           'PROCESS_PLANNING_SCHEMA.PARTIAL_CIRCULAR_PROFILE',
           'PROCESS_PLANNING_SCHEMA.PATH_FEATURE_COMPONENT',
           'PROCESS_PLANNING_SCHEMA.POCKET_BOTTOM',
           'PROCESS_PLANNING_SCHEMA.PROFILE_FLOOR',
           'PROCESS_PLANNING_SCHEMA.RECTANGULAR_CLOSED_PROFILE',
           'PROCESS_PLANNING_SCHEMA.RIB_TOP_FLOOR',
           'PROCESS_PLANNING_SCHEMA.ROUNDED_U_PROFILE',
           'PROCESS_PLANNING_SCHEMA.SLOT_END',
           'PROCESS_PLANNING_SCHEMA.SQUARE_U_PROFILE',
           'PROCESS_PLANNING_SCHEMA.SYMMETRIC_SHAPE_ASPECT',
           'PROCESS_PLANNING_SCHEMA.TAPER',
           'PROCESS_PLANNING_SCHEMA.TEE_PROFILE',
           'PROCESS_PLANNING_SCHEMA.TOLERANCE_ZONE',
           'PROCESS_PLANNING_SCHEMA.TRANSITION_FEATURE',
```

```
            'PROCESS_PLANNING_SCHEMA.VEE_PROFILE'],3)) )) = 0;

  END_RULE; -- subtype_exclusiveness_shape_aspect

  RULE subtype_exclusiveness_shape_representation FOR (
            shape_representation);

  WHERE
    wr1: SIZEOF(QUERY ( a <* shape_representation | (NOT
            type_check_function(a,[
            'PROCESS_PLANNING_SCHEMA.ADVANCED_BREP_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.DIRECTION_SHAPE_REPRESENTATION',
      'PROCESS_PLANNING_SCHEMA.EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.FACE_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.FACETED_BREP_SHAPE_REPRESENTATION',
'PROCESS_PLANNING_SCHEMA.GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION',
'PROCESS_PLANNING_SCHEMA.GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATI
ON',
            'PROCESS_PLANNING_SCHEMA.LOCATION_SHAPE_REPRESENTATION',
          'PROCESS_PLANNING_SCHEMA.MANIFOLD_SURFACE_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.PATH_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.PLANAR_SHAPE_REPRESENTATION',
            'PROCESS_PLANNING_SCHEMA.SHAPE_DIMENSION_REPRESENTATION',
       'PROCESS_PLANNING_SCHEMA.SHAPE_REPRESENTATION_WITH_PARAMETERS',
      'PROCESS_PLANNING_SCHEMA.SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION'],
            3)) )) = 0;

  END_RULE; -- subtype_exclusiveness_shape_representation

  FUNCTION acyclic_curve_replica(
              rep: curve_replica;
              parent: curve
    ): BOOLEAN;
    IF NOT ('PROCESS_PLANNING_SCHEMA.CURVE_REPLICA' IN TYPEOF(parent))
        THEN
      RETURN(TRUE);
    END_IF;
    IF parent :=: rep THEN
      RETURN(FALSE);
    ELSE
      RETURN(acyclic_curve_replica(rep,parent\curve_replica.parent_curve));
    END_IF;

  END_FUNCTION; -- acyclic_curve_replica

  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 | (
        'PROCESS_PLANNING_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],'')) | (
          'PROCESS_PLANNING_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

FUNCTION acyclic_point_replica(
            rep: point_replica;
            parent: point
    ): BOOLEAN;
  IF NOT ('PROCESS_PLANNING_SCHEMA.POINT_REPLICA' IN TYPEOF(parent))
      THEN
    RETURN(TRUE);
  END_IF;
  IF parent :=: rep THEN
    RETURN(FALSE);
  ELSE
    RETURN(acyclic_point_replica(rep,parent\point_replica.parent_pt));
  END_IF;

END_FUNCTION; -- acyclic_point_replica

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.relating_product_definition IN relatives THEN
    RETURN(FALSE);
  END_IF;
  x := QUERY ( pd <* bag_to_set(USEDIN(relation.
      relating_product_definition,('PROCESS_PLANNING_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.relating_product_definition,specific_relation) THEN
      RETURN(FALSE);
    END_IF;
  END_REPEAT;
  RETURN(TRUE);
```

```
END_FUNCTION; -- acyclic_product_definition_relationship

FUNCTION acyclic_surface_replica(
            rep: surface_replica;
            parent: surface
    ): BOOLEAN;
  IF NOT ('PROCESS_PLANNING_SCHEMA.SURFACE_REPLICA' IN TYPEOF(parent))
       THEN
    RETURN(TRUE);
  END_IF;
  IF parent :=: rep THEN
    RETURN(FALSE);
  ELSE
    RETURN(acyclic_surface_replica(rep,parent\surface_replica.
        parent_surface));
  END_IF;

END_FUNCTION; -- acyclic_surface_replica

FUNCTION associated_surface(
            arg: pcurve_or_surface
    ): surface;

  LOCAL
    surf : surface;
  END_LOCAL;
  IF 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(arg) THEN
    surf := arg.basis_surface;
  ELSE
    surf := arg;
  END_IF;
  RETURN(surf);

END_FUNCTION; -- associated_surface

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

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

FUNCTION boolean_choose(
            b: BOOLEAN;
            choice1, choice2: GENERIC:item
    ): GENERIC:item;
  IF b THEN
    RETURN(choice1);
  ELSE
    RETURN(choice2);
  END_IF;

END_FUNCTION; -- boolean_choose

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

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

FUNCTION closed_shell_reversed(
            a_shell: closed_shell
    ): oriented_closed_shell;

  LOCAL
    the_reverse : oriented_closed_shell;
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

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

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 ('PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve))) AND (NOT (
        'PROCESS_PLANNING_SCHEMA.SURFACE_CURVE' IN TYPEOF(c\
        composite_curve.segments[k].parent_curve)))) AND (NOT (
        'PROCESS_PLANNING_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

FUNCTION constraints_geometry_shell_based_surface_model(
            m: shell_based_surface_model
    ): BOOLEAN;

  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT j := 1 TO SIZEOF(m.sbsm_boundary) BY 1;
    IF (NOT ('PROCESS_PLANNING_SCHEMA.OPEN_SHELL' IN TYPEOF(m.
        sbsm_boundary[j]))) AND (NOT (
        'PROCESS_PLANNING_SCHEMA.CLOSED_SHELL' IN TYPEOF(m.sbsm_boundary[
        j]))) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- constraints_geometry_shell_based_surface_model

FUNCTION constraints_geometry_shell_based_wireframe_model(
            m: shell_based_wireframe_model
    ): BOOLEAN;

  LOCAL
    result : BOOLEAN := TRUE;
  END_LOCAL;
  REPEAT j := 1 TO SIZEOF(m.sbwm_boundary) BY 1;
    IF (NOT ('PROCESS_PLANNING_SCHEMA.WIRE_SHELL' IN TYPEOF(m.
        sbwm_boundary[j]))) AND (NOT (
        'PROCESS_PLANNING_SCHEMA.VERTEX_SHELL' IN TYPEOF(m.sbwm_boundary[
        j]))) THEN
      result := FALSE;
      RETURN(result);
    END_IF;
  END_REPEAT;
  RETURN(result);

END_FUNCTION; -- constraints_geometry_shell_based_wireframe_model

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

  FUNCTION constraints_rectangular_composite_surface(
              s: rectangular_composite_surface
      ): BOOLEAN;
    REPEAT i := 1 TO s.n_u BY 1;
      REPEAT j := 1 TO s.n_v BY 1;
        IF NOT (('PROCESS_PLANNING_SCHEMA.B_SPLINE_SURFACE' IN TYPEOF(s.
            segments[i][j].parent_surface)) OR (
            'PROCESS_PLANNING_SCHEMA.RECTANGULAR_TRIMMED_SURFACE' IN
            TYPEOF(s.segments[i][j].parent_surface))) THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_REPEAT;
    REPEAT i := 1 TO s.n_u - 1 BY 1;
      REPEAT j := 1 TO s.n_v BY 1;
        IF s.segments[i][j].u_transition = discontinuous THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_REPEAT;
    REPEAT i := 1 TO s.n_u BY 1;
      REPEAT j := 1 TO s.n_v - 1 BY 1;
        IF s.segments[i][j].v_transition = discontinuous THEN
          RETURN(FALSE);
        END_IF;
      END_REPEAT;
    END_REPEAT;
    RETURN(TRUE);

  END_FUNCTION; -- constraints_rectangular_composite_surface

  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

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

FUNCTION derive_dimensional_exponents(
            x: unit
    ): dimensional_exponents;

  LOCAL
    result : dimensional_exponents := dimensional_exponents(0,0,0,0,0,0,
             0);
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

FUNCTION dimension_of(
            item: geometric_representation_item
    ): dimension_count;

  LOCAL
    x   : SET OF representation;
    y   : representation_context;
    dim : dimension_count;
  END_LOCAL;
  IF 'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(item) THEN
    dim := SIZEOF(item\cartesian_point.coordinates);
    RETURN(dim);
  END_IF;
  IF 'PROCESS_PLANNING_SCHEMA.DIRECTION' IN TYPEOF(item) THEN
    dim := SIZEOF(item\direction.direction_ratios);
    RETURN(dim);
  END_IF;
  IF 'PROCESS_PLANNING_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

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

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

FUNCTION edge_reversed(
            an_edge: edge
```

```
    ): oriented_edge;

  LOCAL
    the_reverse : oriented_edge;
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

FUNCTION face_bound_reversed(
            a_face_bound: face_bound
    ): face_bound;

  LOCAL
    the_reverse : face_bound;
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

FUNCTION face_reversed(
            a_face: face
    ): oriented_face;

  LOCAL
    the_reverse : oriented_face;
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

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
      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

FUNCTION gbsf_check_curve(
            cv: representation_item
    ): BOOLEAN;
  IF SIZEOF(['PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE',
      'PROCESS_PLANNING_SCHEMA.CONIC',
      'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA',
      'PROCESS_PLANNING_SCHEMA.LINE',
      'PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1 THEN
    RETURN(FALSE);
  END_IF;
  IF SIZEOF(['PROCESS_PLANNING_SCHEMA.CIRCLE',
      'PROCESS_PLANNING_SCHEMA.ELLIPSE',
      'PROCESS_PLANNING_SCHEMA.TRIMMED_CURVE'] * TYPEOF(cv)) = 1 THEN
    RETURN(TRUE);
  ELSE
    IF (('PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv
        \b_spline_curve.self_intersect = FALSE)) OR (cv\b_spline_curve.
        self_intersect = UNKNOWN) THEN
      RETURN(TRUE);
    ELSE
      IF (('PROCESS_PLANNING_SCHEMA.COMPOSITE_CURVE' IN TYPEOF(cv)) AND
          (cv\composite_curve.self_intersect = FALSE)) OR (cv\
          composite_curve.self_intersect = UNKNOWN) THEN
        RETURN(SIZEOF(QUERY ( seg <* cv\composite_curve.segments | (NOT
            gbsf_check_curve(seg.parent_curve)) )) = 0);
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA' IN TYPEOF(cv) THEN
          RETURN(gbsf_check_curve(cv\curve_replica.parent_curve));
```

```
        ELSE
          IF (('PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF(cv))
              AND ((cv\offset_curve_3d.self_intersect = FALSE) OR (cv\
              offset_curve_3d.self_intersect = UNKNOWN))) AND (NOT (
              'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(cv\
              offset_curve_3d.basis_curve))) THEN
            RETURN(gbsf_check_curve(cv\offset_curve_3d.basis_curve));
          ELSE
            IF 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(cv) THEN
              RETURN(gbsf_check_curve(cv\pcurve.reference_to_curve\
                  representation.items[1]) AND gbsf_check_surface(cv\
                  pcurve.basis_surface));
            ELSE
              IF 'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(cv) THEN
                IF SIZEOF(cv\polyline.points) >= 3 THEN
                  RETURN(TRUE);
                END_IF;
              ELSE
                IF 'PROCESS_PLANNING_SCHEMA.SURFACE_CURVE' IN TYPEOF(cv)
                     THEN
                  IF gbsf_check_curve(cv\surface_curve.curve_3d) THEN
                    REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                        associated_geometry) BY 1;
                      IF 'PROCESS_PLANNING_SCHEMA.SURFACE' IN TYPEOF(cv\
                          surface_curve.associated_geometry[i]) THEN
                        IF NOT gbsf_check_surface(cv\surface_curve.
                            associated_geometry[i]) THEN
                          RETURN(FALSE);
                        END_IF;
                      ELSE
                        IF 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(cv
                            \surface_curve.associated_geometry[i]) THEN
                          IF NOT gbsf_check_curve(cv\surface_curve.
                              associated_geometry[i]) THEN
                            RETURN(FALSE);
                          END_IF;
                        END_IF;
                      END_IF;
                    END_REPEAT;
                    RETURN(TRUE);
                  END_IF;
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- gbsf_check_curve

FUNCTION gbsf_check_point(
             pnt: point
    ): BOOLEAN;
  IF 'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
    RETURN(TRUE);
  ELSE
    IF 'PROCESS_PLANNING_SCHEMA.POINT_ON_CURVE' IN TYPEOF(pnt) THEN
      RETURN(gbsf_check_curve(pnt\point_on_curve.basis_curve));
```

 

```
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.POINT_ON_SURFACE' IN TYPEOF(pnt) THEN
        RETURN(gbsf_check_surface(pnt\point_on_surface.basis_surface));
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.DEGENERATE_PCURVE' IN TYPEOF(pnt)
             THEN
          RETURN(gbsf_check_curve(pnt\degenerate_pcurve.
              reference_to_curve\representation.items[1]) AND
              gbsf_check_surface(pnt\degenerate_pcurve.basis_surface));
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- gbsf_check_point

FUNCTION gbsf_check_surface(
            sf: surface
    ): BOOLEAN;
  IF (('PROCESS_PLANNING_SCHEMA.B_SPLINE_SURFACE' IN TYPEOF(sf)) AND (sf
      \b_spline_surface.self_intersect = FALSE)) OR (sf\b_spline_surface.
      self_intersect = UNKNOWN) THEN
    RETURN(TRUE);
  ELSE
    IF SIZEOF(['PROCESS_PLANNING_SCHEMA.SPHERICAL_SURFACE',
        'PROCESS_PLANNING_SCHEMA.TOROIDAL_SURFACE',
        'PROCESS_PLANNING_SCHEMA.CURVE_BOUNDED_SURFACE',
        'PROCESS_PLANNING_SCHEMA.RECTANGULAR_TRIMMED_SURFACE'] * TYPEOF(
        sf)) = 1 THEN
      RETURN(TRUE);
    ELSE
      IF (('PROCESS_PLANNING_SCHEMA.OFFSET_SURFACE' IN TYPEOF(sf)) AND (
          sf\offset_surface.self_intersect = FALSE)) OR (sf\
          offset_surface.self_intersect = UNKNOWN) THEN
        RETURN(gbsf_check_surface(sf\offset_surface.basis_surface));
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.RECTANGULAR_COMPOSITE_SURFACE' IN
            TYPEOF(sf) THEN
          REPEAT i := 1 TO SIZEOF(sf\rectangular_composite_surface.
              segments) BY 1;
            REPEAT j := 1 TO SIZEOF(sf\rectangular_composite_surface.
                segments[i]) BY 1;
              IF NOT gbsf_check_surface(sf\rectangular_composite_surface
                  .segments[i][j].parent_surface) THEN
                RETURN(FALSE);
              END_IF;
            END_REPEAT;
          END_REPEAT;
          RETURN(TRUE);
        ELSE
          IF 'PROCESS_PLANNING_SCHEMA.SURFACE_REPLICA' IN TYPEOF(sf)
               THEN
            RETURN(gbsf_check_surface(sf\surface_replica.parent_surface));
          ELSE
            IF 'PROCESS_PLANNING_SCHEMA.SURFACE_OF_REVOLUTION' IN
                TYPEOF(sf) THEN
              RETURN(gbsf_check_curve(sf\swept_surface.swept_curve));
            END_IF;
          END_IF;
        END_IF;
```

```
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- gbsf_check_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 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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

FUNCTION get_description_value(
            obj: description_attribute_select
    ): text;

  LOCAL
    description_bag : BAG OF description_attribute := USEDIN(obj,(
                        'PROCESS_PLANNING_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

FUNCTION get_id_value(
```

 

```
              obj: id_attribute_select
    ): identifier;

  LOCAL
    id_bag : BAG OF id_attribute := USEDIN(obj,(
              'PROCESS_PLANNING_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

FUNCTION get_name_value(
              obj: name_attribute_select
    ): label;

  LOCAL
    name_bag : BAG OF name_attribute := USEDIN(obj,(
                'PROCESS_PLANNING_SCHEMA.' + 'NAME_ATTRIBUTE.') +
                'NAMED_ITEM');
  END_LOCAL;
  IF SIZEOF(name_bag) = 1 THEN
    RETURN(name_bag[1].attribute_value);
  ELSE
    RETURN(?);
  END_IF;

END_FUNCTION; -- get_name_value

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,
      'PROCESS_PLANNING_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],
'PROCESS_PLANNING_SCHEMA.PROPERTY_DEFINITION_REPRESENTATION.DEFINITION'));
  END_REPEAT;
  RETURN(pdr_set);

END_FUNCTION; -- get_property_definition_representations

FUNCTION get_role(
              obj: role_select
    ): object_role;

  LOCAL
    role_bag : BAG OF role_association := USEDIN(obj,(
                'PROCESS_PLANNING_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

FUNCTION item_in_context(
             item: representation_item;
             cntxt: representation_context
    ): BOOLEAN;

  LOCAL
    y : BAG OF representation_item;
  END_LOCAL;
  IF SIZEOF(USEDIN(item,'PROCESS_PLANNING_SCHEMA.REPRESENTATION.ITEMS')
      * cntxt.representations_in_context) > 0 THEN
    RETURN(TRUE);
  ELSE
    y := QUERY ( z <* USEDIN(item,'') | (
        'PROCESS_PLANNING_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;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- item_in_context

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

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
```

 

```
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

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

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

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

  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 := 'PROCESS_PLANNING_SCHEMA.POLY_LOOP' IN TYPEOF(l[1]);
    REPEAT i := 2 TO SIZEOF(l) BY 1;
      IF ('PROCESS_PLANNING_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

  FUNCTION msb_shells(
              brep: manifold_solid_brep
      ): SET [1:?] OF closed_shell;
    IF SIZEOF(QUERY ( msbtype <* TYPEOF(brep) | (msbtype LIKE
        '*BREP_WITH_VOIDS') )) >= 1 THEN
      RETURN(brep\brep_with_voids.voids + brep.outer);
    ELSE
      RETURN([brep.outer]);
    END_IF;

  END_FUNCTION; -- msb_shells

  FUNCTION msf_curve_check(
              cv: representation_item
      ): BOOLEAN;
    IF SIZEOF(['PROCESS_PLANNING_SCHEMA.BOUNDED_CURVE',
        'PROCESS_PLANNING_SCHEMA.CONIC',
        'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA',
        'PROCESS_PLANNING_SCHEMA.LINE',
        'PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D'] * TYPEOF(cv)) > 1 THEN
      RETURN(FALSE);
    END_IF;
    IF (('PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE' IN TYPEOF(cv)) AND (cv\
        b_spline_curve.self_intersect = FALSE)) OR (cv\b_spline_curve.
        self_intersect = UNKNOWN) THEN
```

 

```
    RETURN(TRUE);
  ELSE
    IF SIZEOF(['PROCESS_PLANNING_SCHEMA.CONIC',
        'PROCESS_PLANNING_SCHEMA.LINE'] * TYPEOF(cv)) = 1 THEN
      RETURN(TRUE);
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA' IN TYPEOF(cv) THEN
        RETURN(msf_curve_check(cv\curve_replica.parent_curve));
      ELSE
        IF (('PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF(cv))
            AND ((cv\offset_curve_3d.self_intersect = FALSE) OR (cv\
            offset_curve_3d.self_intersect = UNKNOWN))) AND (NOT (
            'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(cv\
            offset_curve_3d.basis_curve))) THEN
          RETURN(msf_curve_check(cv\offset_curve_3d.basis_curve));
        ELSE
          IF 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(cv) THEN
            RETURN(msf_curve_check(cv\pcurve.reference_to_curve\
                representation.items[1]) AND msf_surface_check(cv\pcurve.
                basis_surface));
          ELSE
            IF 'PROCESS_PLANNING_SCHEMA.SURFACE_CURVE' IN TYPEOF(cv)
                THEN
              IF msf_curve_check(cv\surface_curve.curve_3d) THEN
                REPEAT i := 1 TO SIZEOF(cv\surface_curve.
                    associated_geometry) BY 1;
                  IF 'PROCESS_PLANNING_SCHEMA.SURFACE' IN TYPEOF(cv\
                      surface_curve.associated_geometry[i]) THEN
                    IF NOT msf_surface_check(cv\surface_curve.
                        associated_geometry[i]) THEN
                      RETURN(FALSE);
                    END_IF;
                  ELSE
                    IF 'PROCESS_PLANNING_SCHEMA.PCURVE' IN TYPEOF(cv\
                        surface_curve.associated_geometry[i]) THEN
                      IF NOT msf_curve_check(cv\surface_curve.
                          associated_geometry[i]) THEN
                        RETURN(FALSE);
                      END_IF;
                    END_IF;
                  END_IF;
                END_REPEAT;
                RETURN(TRUE);
              END_IF;
            ELSE
              IF 'PROCESS_PLANNING_SCHEMA.POLYLINE' IN TYPEOF(cv) THEN
                IF SIZEOF(cv\polyline.points) >= 3 THEN
                  RETURN(TRUE);
                END_IF;
              END_IF;
            END_IF;
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- msf_curve_check

FUNCTION msf_surface_check(
```

```
              surf: surface
    ): BOOLEAN;
  IF 'PROCESS_PLANNING_SCHEMA.ELEMENTARY_SURFACE' IN TYPEOF(surf) THEN
    RETURN(TRUE);
  ELSE
    IF 'PROCESS_PLANNING_SCHEMA.SWEPT_SURFACE' IN TYPEOF(surf) THEN
      RETURN(msf_curve_check(surf\swept_surface.swept_curve));
    ELSE
      IF (('PROCESS_PLANNING_SCHEMA.OFFSET_SURFACE' IN TYPEOF(surf)) AND
           (surf\offset_surface.self_intersect = FALSE)) OR (surf\
          offset_surface.self_intersect = UNKNOWN) THEN
        RETURN(msf_surface_check(surf\offset_surface.basis_surface));
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.SURFACE_REPLICA' IN TYPEOF(surf)
             THEN
          RETURN(msf_surface_check(surf\surface_replica.parent_surface));
        ELSE
          IF (('PROCESS_PLANNING_SCHEMA.B_SPLINE_SURFACE' IN TYPEOF(surf))
              AND (surf\b_spline_surface.self_intersect = FALSE)) OR (
              surf\b_spline_surface.self_intersect = UNKNOWN) THEN
            RETURN(TRUE);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- msf_surface_check

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 'PROCESS_PLANNING_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;
      IF 'PROCESS_PLANNING_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

FUNCTION open_shell_reversed(
            a_shell: open_shell
    ): oriented_open_shell;

  LOCAL
    the_reverse : oriented_open_shell;
  END_LOCAL;
  IF 'PROCESS_PLANNING_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

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

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

  FUNCTION path_reversed(
              a_path: path
      ): oriented_path;

    LOCAL
      the_reverse : oriented_path;
    END_LOCAL;
    IF 'PROCESS_PLANNING_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

  FUNCTION scalar_times_vector(
              scalar: REAL;
              vec: vector_or_direction
      ): vector;

    LOCAL
      v      : direction;
      mag    : REAL;
      result : vector;
    END_LOCAL;
    IF (NOT EXISTS(scalar)) OR (NOT EXISTS(vec)) THEN
      RETURN(?);
    ELSE
      IF 'PROCESS_PLANNING_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

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

FUNCTION set_of_topology_reversed(
            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

FUNCTION shell_reversed(
            a_shell: shell
    ): shell;
  IF 'PROCESS_PLANNING_SCHEMA.OPEN_SHELL' IN TYPEOF(a_shell) THEN
    RETURN(open_shell_reversed(a_shell));
  ELSE
    IF 'PROCESS_PLANNING_SCHEMA.CLOSED_SHELL' IN TYPEOF(a_shell) THEN
      RETURN(closed_shell_reversed(a_shell));
    ELSE
      RETURN(?);
    END_IF;
  END_IF;

END_FUNCTION; -- shell_reversed

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

  FUNCTION topology_reversed(
              an_item: reversible_topology
      ): reversible_topology;
    IF 'PROCESS_PLANNING_SCHEMA.EDGE' IN TYPEOF(an_item) THEN
      RETURN(edge_reversed(an_item));
    END_IF;
    IF 'PROCESS_PLANNING_SCHEMA.PATH' IN TYPEOF(an_item) THEN
      RETURN(path_reversed(an_item));
    END_IF;
    IF 'PROCESS_PLANNING_SCHEMA.FACE_BOUND' IN TYPEOF(an_item) THEN
      RETURN(face_bound_reversed(an_item));
    END_IF;
    IF 'PROCESS_PLANNING_SCHEMA.FACE' IN TYPEOF(an_item) THEN
      RETURN(face_reversed(an_item));
    END_IF;
    IF 'PROCESS_PLANNING_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

  FUNCTION type_check_function(
              the_type: GENERIC;
              sub_names: SET OF STRING;
              criterion: INTEGER
      ): LOGICAL;
    IF ((NOT EXISTS(the_type)) OR (NOT ((0 <= criterion) AND (criterion <=
         3)))) OR (SIZEOF(sub_names) = 0) THEN
      RETURN(UNKNOWN);
    ELSE
      CASE criterion OF
        0  :           RETURN(SIZEOF(sub_names * TYPEOF(the_type)) > 0);
        1  :           RETURN(SIZEOF(sub_names * TYPEOF(the_type)) = 0);
        2  :           RETURN(SIZEOF(sub_names * TYPEOF(the_type)) = 1);
        3  :           RETURN(SIZEOF(sub_names * TYPEOF(the_type)) <= 1);
        END_CASE;
    END_IF;

  END_FUNCTION; -- type_check_function

  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,'')) | ((
      'PROCESS_PLANNING_SCHEMA.REPRESENTATION_ITEM' IN TYPEOF(z)) OR (
      'PROCESS_PLANNING_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;
  RETURN(result_items);

END_FUNCTION; -- using_items

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,
      'PROCESS_PLANNING_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],
          'PROCESS_PLANNING_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

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;
    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

FUNCTION valid_geometrically_bounded_wf_curve(
            crv: curve
    ): BOOLEAN;
  IF SIZEOF(['PROCESS_PLANNING_SCHEMA.POLYLINE',
      'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE',
       'PROCESS_PLANNING_SCHEMA.ELLIPSE','PROCESS_PLANNING_SCHEMA.CIRCLE']
      * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
  ELSE
    IF 'PROCESS_PLANNING_SCHEMA.TRIMMED_CURVE' IN TYPEOF(crv) THEN
      IF SIZEOF(['PROCESS_PLANNING_SCHEMA.LINE',
          'PROCESS_PLANNING_SCHEMA.PARABOLA',
          'PROCESS_PLANNING_SCHEMA.HYPERBOLA'] * TYPEOF(crv\trimmed_curve
          .basis_curve)) = 1 THEN
        RETURN(TRUE);
      ELSE
        RETURN(valid_geometrically_bounded_wf_curve(crv\trimmed_curve.
            basis_curve));
      END_IF;
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF(crv) THEN
        RETURN(valid_geometrically_bounded_wf_curve(crv\offset_curve_3d.
            basis_curve));
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA' IN TYPEOF(crv) THEN
```

 

```
          RETURN(valid_geometrically_bounded_wf_curve(crv\curve_replica.
              parent_curve));
        ELSE
          IF 'PROCESS_PLANNING_SCHEMA.COMPOSITE_CURVE' IN TYPEOF(crv)
               THEN
            RETURN(SIZEOF(QUERY ( ccs <* crv\composite_curve.segments |
                (NOT valid_geometrically_bounded_wf_curve(ccs.
                parent_curve)) )) = 0);
          END_IF;
        END_IF;
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_curve

FUNCTION valid_geometrically_bounded_wf_point(
             pnt: point
    ): BOOLEAN;
  IF 'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
    RETURN(TRUE);
  ELSE
    IF 'PROCESS_PLANNING_SCHEMA.POINT_ON_CURVE' IN TYPEOF(pnt) THEN
      RETURN(valid_geometrically_bounded_wf_curve(pnt\point_on_curve.
          basis_curve));
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.POINT_REPLICA' IN TYPEOF(pnt) THEN
        RETURN(valid_geometrically_bounded_wf_point(pnt\point_replica.
            parent_pt));
      END_IF;
    END_IF;
  END_IF;
  RETURN(FALSE);

END_FUNCTION; -- valid_geometrically_bounded_wf_point

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

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

FUNCTION valid_units(
            m: measure_with_unit
    ): BOOLEAN;
  IF 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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 'PROCESS_PLANNING_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) THEN
      RETURN(FALSE);
    END_IF;
  END_IF;
  RETURN(TRUE);

END_FUNCTION; -- valid_units

FUNCTION valid_wireframe_edge_curve(
            crv: curve
    ): BOOLEAN;
  IF SIZEOF(['PROCESS_PLANNING_SCHEMA.LINE',
      'PROCESS_PLANNING_SCHEMA.CONIC',
      'PROCESS_PLANNING_SCHEMA.B_SPLINE_CURVE',
      'PROCESS_PLANNING_SCHEMA.POLYLINE'] * TYPEOF(crv)) = 1 THEN
    RETURN(TRUE);
```

```
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.CURVE_REPLICA' IN TYPEOF(crv) THEN
        RETURN(valid_wireframe_edge_curve(crv\curve_replica.parent_curve));
      ELSE
        IF 'PROCESS_PLANNING_SCHEMA.OFFSET_CURVE_3D' IN TYPEOF(crv) THEN
          RETURN(valid_wireframe_edge_curve(crv\offset_curve_3d.
              basis_curve));
        END_IF;
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- valid_wireframe_edge_curve

  FUNCTION valid_wireframe_vertex_point(
              pnt: point
      ): BOOLEAN;
    IF 'PROCESS_PLANNING_SCHEMA.CARTESIAN_POINT' IN TYPEOF(pnt) THEN
      RETURN(TRUE);
    ELSE
      IF 'PROCESS_PLANNING_SCHEMA.POINT_REPLICA' IN TYPEOF(pnt) THEN
        RETURN(valid_wireframe_vertex_point(pnt\point_replica.parent_pt));
      END_IF;
    END_IF;
    RETURN(FALSE);

  END_FUNCTION; -- valid_wireframe_vertex_point

  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 'PROCESS_PLANNING_SCHEMA.VECTOR' IN TYPEOF(arg1) THEN
          mag1 := arg1.magnitude;
          vec1 := arg1.orientation;
        ELSE
          mag1 := 1;
          vec1 := arg1;
        END_IF;
        IF 'PROCESS_PLANNING_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;
        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

END_SCHEMA; -- process_planning_schema
```

# Annex B
(normative)

# AIM short names

The following table provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of short names are found in the implementation methods included in ISO 10303.

## Table 2 — AIM short names

| | |
|---|---|
| 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_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 |
| ADDRESS | ADDRSS |
| ADVANCED_BREP_SHAPE_REPRESENTATION | ABSR |
| ADVANCED_FACE | ADVFC |
| ALLOWED_TIME | ALLTM |
| ALTERNATE_ACTION_METHOD_RELATIONSHIP | AAMR |
| ALTERNATE_PLAN_RELATIONSHIP | ALPLRL |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| 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_CONTRACT_ASSIGNMENT | APCNAS |
| 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_SECURITY_CLASSIFICATION_ASSIGNMENT | ASCA |
| APPROVAL | APPRVL |
| APPROVAL_ASSIGNMENT | APPASS |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| APPROVAL_DATE_TIME | APDTTM |
| APPROVAL_PERSON_ORGANIZATION | APPROR |
| APPROVAL_ROLE | APPRL |
| APPROVAL_STATUS | APPSTT |
| 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 |
| BEZIER_CURVE | BZRCRV |
| BEZIER_SURFACE | BZRSRF |
| BLOCK_SHAPE_REPRESENTATION | BLSHRP |
| BOSS | BOSS |
| BOSS_TOP | BSSTP |
| BOUNDARY_CURVE | BNDCR |
| BOUNDED_CURVE | BNDCRV |
| BOUNDED_PCURVE | BNDPCR |
| BOUNDED_SURFACE | BNDSRF |
| BOUNDED_SURFACE_CURVE | BNSRCR |
| BREP_WITH_VOIDS | BRWTVD |
| CALENDAR_DATE | CLNDT |
| CARTESIAN_POINT | CRTPNT |
| CARTESIAN_TRANSFORMATION_OPERATOR | CRTROP |
| CARTESIAN_TRANSFORMATION_OPERATOR_3D | CTO3 |
| CENTRE_OF_SYMMETRY | CNOFSY |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| CHAMFER | CHMFR |
| CHAMFER_OFFSET | CHMOFF |
| CHARACTERIZED_OBJECT | CHROBJ |
| CIRCLE | CIRCLE |
| CIRCULAR_CLOSED_PROFILE | CRCLPR |
| CIRCULAR_PATTERN | CRCPTT |
| CIRCULAR_RUNOUT_TOLERANCE | CRRNTL |
| CLASS | CLASS |
| CLASSIFICATION_ASSIGNMENT | CLSASS |
| CLASSIFICATION_ROLE | CLSRL |
| CLOSED_PATH_PROFILE | CLPTPR |
| CLOSED_SHELL | CLSSHL |
| COAXIALITY_TOLERANCE | CXLTLR |
| COMMON_DATUM | CMMDTM |
| COMPOSITE_CURVE | CMPCRV |
| COMPOSITE_CURVE_ON_SURFACE | CCOS |
| COMPOSITE_CURVE_SEGMENT | CMCRSG |
| COMPOSITE_HOLE | CMPHL |
| COMPOSITE_SHAPE_ASPECT | CMSHAS |
| COMPOUND_FEATURE | CMPFTR |
| COMPOUND_REPRESENTATION_ITEM | CMRPIT |
| CONCENTRICITY_TOLERANCE | CNCTLR |
| CONIC | CONIC |
| CONICAL_SURFACE | CNCSRF |
| CONNECTED_EDGE_SET | CNEDST |
| CONNECTED_FACE_SET | CNFCST |
| CONTEXT_DEPENDENT_UNIT | CNDPUN |
| CONTINUOUS_PROCESS_RELATIONSHIP | CNPRRL |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| CONTRACT | CNTRCT |
| CONTRACT_ASSIGNMENT | CNTASS |
| CONTRACT_TYPE | CNTTYP |
| CONTROLLER | CNTRLL |
| CONTROLLER_PROGRAM | CNTPRG |
| CONVERSION_BASED_UNIT | CNBSUN |
| COORDINATED_UNIVERSAL_TIME_OFFSET | CUTO |
| CURVE | CURVE |
| CURVE_BOUNDED_SURFACE | CRBNSR |
| CURVE_REPLICA | CRVRPL |
| CYLINDRICAL_SHAPE_REPRESENTATION | CYSHRP |
| CYLINDRICAL_SURFACE | CYLSRF |
| CYLINDRICITY_TOLERANCE | CYLTLR |
| DATA_ENVIRONMENT | DTENV |
| DATE | DATE |
| DATE_AND_TIME | DTANTM |
| DATE_ASSIGNMENT | DTASS |
| DATE_ROLE | DTRL |
| DATUM | DATUM |
| DATUM_FEATURE | DTMFTR |
| DATUM_REFERENCE | DTMRFR |
| DATUM_TARGET | DTMTRG |
| DEFINING_ACTION_METHOD_RELATIONSHIP | DAMR |
| DEFINITIONAL_REPRESENTATION | DFNRPR |
| DEGENERATE_PCURVE | DGNPCR |
| DEGENERATE_TOROIDAL_SURFACE | DGTRSR |
| DERIVED_SHAPE_ASPECT | DRSHAS |
| DERIVED_UNIT | DRVUNT |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| DERIVED_UNIT_ELEMENT | DRUNEL |
| DESCRIPTION_ATTRIBUTE | DSCATT |
| DESCRIPTIVE_REPRESENTATION_ITEM | DSRPIT |
| DESIGN_REFERENCE | DSGRFR |
| DIMENSION_RELATED_TOLERANCE_ZONE_ELEMENT | DRTZE |
| DIMENSIONAL_CHARACTERISTIC_REPRESENTATION | DMCHRP |
| DIMENSIONAL_EXPONENTS | DMNEXP |
| DIMENSIONAL_LOCATION | DMNLCT |
| DIMENSIONAL_LOCATION_WITH_PATH | DLWP |
| 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 | DCP1 |
| 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 |
| EDGE | EDGE |
| EDGE_BASED_WIREFRAME_MODEL | EBWM |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| EDGE_BASED_WIREFRAME_SHAPE_REPRESENTATION | EBWSR |
| EDGE_CURVE | EDGCRV |
| EDGE_LOOP | EDGLP |
| EDGE_ROUND | EDGRND |
| EFFECTIVITY | EFFCTV |
| EFFECTIVITY_ASSIGNMENT | EFFASS |
| ELEMENTARY_SURFACE | ELMSRF |
| ELLIPSE | ELLPS |
| EVALUATED_DEGENERATE_PCURVE | EVDGPC |
| EXECUTED_ACTION | EXCACT |
| EXTENSION | EXTNSN |
| EXTERNAL_IDENTIFICATION_ASSIGNMENT | EXIDAS |
| EXTERNAL_SOURCE | EXTSRC |
| 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 |
| FACE | FACE |
| FACE_BOUND | FCBND |
| FACE_OUTER_BOUND | FCOTBN |
| FACE_SHAPE_REPRESENTATION | FCSHRP |
| FACE_SURFACE | FCSRF |
| FACETED_BREP | FCTBR |
| FACETED_BREP_SHAPE_REPRESENTATION | FBSR |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| FEATURE_COMPONENT_DEFINITION | FTCMDF |
| FEATURE_COMPONENT_RELATIONSHIP | FTCMRL |
| FEATURE_DEFINITION | FTRDFN |
| FEATURE_DEPENDENCY | FTRDPN |
| FEATURE_IDENTIFICATION_ITEM | FTIDIT |
| FEATURE_INTERACTION | FTRINT |
| FEATURE_PATTERN | FTRPTT |
| FEATURE_PROCESS | FTRPRC |
| FILLET | FILLET |
| FIXTURE_ASSEMBLY | FXTASS |
| FIXTURE_ASSEMBLY_ELEMENT | FXASEL |
| FIXTURE_ASSEMBLY_RELATIONSHIP | FXASRL |
| FIXTURE_CONTRACT | FXTCNT |
| FIXTURE_MACHINE_MOUNTING | FXMCMN |
| FIXTURE_MACHINE_UNMOUNTING | FXMCUN |
| FIXTURE_PALLET_MOUNTING | FXPLMN |
| FIXTURE_PALLET_UNMOUNTING | FXPLUN |
| FIXTURE_SETUP | FXTSTP |
| FLAT_FACE | FLTFC |
| FLATNESS_TOLERANCE | FLTTLR |
| FOUNDED_ITEM | FNDITM |
| FUNCTIONALLY_DEFINED_TRANSFORMATION | FNDFTR |
| GEAR | GEAR |
| GENERAL_PROPERTY | GNRPRP |
| GENERAL_PROPERTY_ASSOCIATION | GNPRAS |
| GENERIC_MANUFACTURING_RESOURCE | GNMNRS |
| GEOMETRIC_ALIGNMENT | GMTALG |
| GEOMETRIC_CURVE_SET | GMCRST |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| GEOMETRIC_INTERSECTION | GMTINT |
| GEOMETRIC_REPRESENTATION_CONTEXT | GMRPCN |
| GEOMETRIC_REPRESENTATION_ITEM | GMRPIT |
| GEOMETRIC_SET | GMTST |
| GEOMETRIC_TOLERANCE | GMTTLR |
| GEOMETRIC_TOLERANCE_RELATIONSHIP | GMTLRL |
| GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE | GTWDR |
| GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT | GTWDU |
| GEOMETRICALLY_BOUNDED_SURFACE_SHAPE_REPRESENTATION | GBSSR |
| GEOMETRICALLY_BOUNDED_WIREFRAME_SHAPE_REPRESENTATION | GBWSR |
| GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT | GC |
| GLOBAL_UNIT_ASSIGNED_CONTEXT | GUAC |
| GROUP | GROUP |
| GROUP_ASSIGNMENT | GRPASS |
| GROUP_RELATIONSHIP | GRPRLT |
| HOLE_BOTTOM | HLBTT |
| HYPERBOLA | HYPRBL |
| ID_ATTRIBUTE | IDATT |
| IDENTIFICATION_ASSIGNMENT | IDNASS |
| IDENTIFICATION_ROLE | IDNRL |
| ILLUSTRATION | ILLSTR |
| IN_FACILITY_LOCATION | INFCLC |
| INSTANCED_FEATURE | INSFTR |
| INTERSECTION_CURVE | INTCRV |
| KNOWN_SOURCE | KNWSRC |
| LENGTH_MEASURE_WITH_UNIT | LMWU |
| LENGTH_UNIT | LNGUNT |
| LIBRARY_CLASS_VERSION_ASSIGNMENT | LCVA |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| 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 |
| MACHINE | MCHN |
| MACHINE_ELEMENT_RELATIONSHIP | MCELRL |
| MACHINE_SETUP | MCHSTP |
| MACHINE_USAGE | MCHUSG |
| MACHINING_PROCESS | MCHPRC |
| MACHINING_TOLERANCE | MCHTLR |
| MAKE_FROM_USAGE_OPTION | MFUO |
| MANIFOLD_SOLID_BREP | MNSLBR |
| MANIFOLD_SURFACE_SHAPE_REPRESENTATION | MSSR |
| 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 |
| MATERIAL_DESIGNATION | MTRDSG |
| MATERIAL_PROPERTY | MTRPRP |
| MATERIAL_PROPERTY_REPRESENTATION | MTPRRP |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| MEASURE_QUALIFICATION | MSRQLF |
| MEASURE_REPRESENTATION_ITEM | MSRPIT |
| MEASURE_WITH_UNIT | MSWTUN |
| MODIFIED_GEOMETRIC_TOLERANCE | MDGMTL |
| MODIFIED_PATTERN | MDFPTT |
| MOUNTING_POSITION | MNTPST |
| NAME_ATTRIBUTE | NMATT |
| NAMED_UNIT | NMDUNT |
| NEXT_ASSEMBLY_USAGE_OCCURRENCE | NAUO |
| NGON_CLOSED_PROFILE | NGCLPR |
| NGON_SHAPE_REPRESENTATION | NGSHRP |
| NON_MACHINING_PROCESS | NNMCPR |
| OBJECT_ROLE | OBJRL |
| OFFSET_CURVE_3D | OFCR3D |
| OFFSET_SURFACE | OFFSRF |
| OPEN_PATH_PROFILE | OPPTPR |
| OPEN_SHELL | OPNSHL |
| 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 |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| OUTER_BOUNDARY_CURVE | OTBNCR |
| OUTER_ROUND | OTRRND |
| OUTSIDE_PROFILE | OTSPRF |
| PALLET | PALLET |
| PALLET_MACHINE_MOUNTING | PLMCMN |
| PARABOLA | PRBL |
| PARALLEL_OFFSET | PRLOFF |
| PARALLELISM_TOLERANCE | PRLTLR |
| PARAMETRIC_REPRESENTATION_CONTEXT | PRRPCN |
| PART_CONTRACT | PRTCNT |
| PART_FIXTURE_MOUNTING | PRFXMN |
| PART_HOLDING_POSITION | PRHLPS |
| PART_MACHINE_MOUNTING | PRMCMN |
| PART_MOUNTING | PRTMNT |
| PART_ROUTING | PRTRTN |
| PART_UNMOUNTING | PRTUNM |
| PARTIAL_CIRCULAR_PROFILE | PRCRPR |
| PATH | PATH |
| PATH_FEATURE_COMPONENT | PTFTCM |
| PATH_SHAPE_REPRESENTATION | PTSHRP |
| PATTERN_OFFSET_MEMBERSHIP | PTOFMM |
| PATTERN_OMIT_MEMBERSHIP | PTOMMM |
| PCURVE | PCURVE |
| PERPENDICULAR_TO | PRPT |
| PERPENDICULARITY_TOLERANCE | PRPTLR |
| PERSON | PERSON |
| PERSON_AND_ORGANIZATION | PRANOR |
| PERSONAL_ADDRESS | PRSADD |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| PLACEMENT | PLCMNT |
| PLANAR_SHAPE_REPRESENTATION | PLSHRP |
| PLANE | PLANE |
| PLANE_ANGLE_MEASURE_WITH_UNIT | PAMWU |
| PLANE_ANGLE_UNIT | PLANUN |
| PLUS_MINUS_TOLERANCE | PLMNTL |
| POCKET | POCKET |
| POCKET_BOTTOM | PCKBTT |
| POINT | POINT |
| POINT_ON_CURVE | PNONCR |
| POINT_ON_SURFACE | PNONSR |
| POINT_REPLICA | PNTRPL |
| POLY_LOOP | PLYLP |
| POLYLINE | PLYLN |
| POSITION_TOLERANCE | PSTTLR |
| PRE_DEFINED_ITEM | PRDFIT |
| PRECISION_QUALIFIER | PRCQLF |
| PROCESS_ACTIVITY | PRCACT |
| PROCESS_PLAN_ACTIVITY | PRPLAC |
| PROCESS_PLAN_SECURITY | PRPLSC |
| PROCESS_PLAN_SPECIFICATION | PRPLSP |
| PROCESS_PLAN_VERSION | PRPLVR |
| PROCESS_PRODUCT_ASSOCIATION | PRPRAS |
| PROCESS_PROPERTY_ASSOCIATION | PRPRS |
| PRODUCT | PRDCT |
| PRODUCT_CATEGORY | PRDCTG |
| PRODUCT_CONTEXT | PRDCNT |
| PRODUCT_DEFINITION | PRDDFN |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| PRODUCT_DEFINITION_CONTEXT | PRDFCN |
| PRODUCT_DEFINITION_EFFECTIVITY | PRDFEF |
| PRODUCT_DEFINITION_FORMATION | PRDFFR |
| PRODUCT_DEFINITION_PROCESS | PRDFPR |
| PRODUCT_DEFINITION_RELATIONSHIP | PRDFRL |
| PRODUCT_DEFINITION_SHAPE | PRDFSH |
| PRODUCT_DEFINITION_USAGE | PRDFUS |
| PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS | PDWAD |
| PRODUCT_RELATED_PRODUCT_CATEGORY | PRPC |
| PRODUCTION_RATE | PRDRT |
| PROFILE_FLOOR | PRFFLR |
| PROJECTED_ZONE_DEFINITION | PRZNDF |
| PROPERTY_DEFINITION | PRPDFN |
| PROPERTY_DEFINITION_RELATIONSHIP | PRDFR |
| PROPERTY_DEFINITION_REPRESENTATION | PRDFRP |
| PROPERTY_PROCESS | PRPPRC |
| PROTRUSION | PRTRSN |
| QUALIFIED_REPRESENTATION_ITEM | QLRPIT |
| QUASI_UNIFORM_CURVE | QSUNCR |
| QUASI_UNIFORM_SURFACE | QSUNSR |
| RANGE_OF_PARTS | RNOFPR |
| RATIO_MEASURE_WITH_UNIT | RMWU |
| RATIO_UNIT | RTUNT |
| RATIONAL_B_SPLINE_CURVE | RBSC |
| RATIONAL_B_SPLINE_SURFACE | RBSS |
| RECTANGULAR_CLOSED_PROFILE | RCCLPR |
| RECTANGULAR_COMPOSITE_SURFACE | RCCMSR |
| RECTANGULAR_PATTERN | RCTPTT |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| RECTANGULAR_TRIMMED_SURFACE | RCTRSR |
| REFERENCED_MODIFIED_DATUM | RFMDDT |
| RELATIONSHIP_CONDITION | RLTCND |
| REMOVAL_VOLUME | RMVVLM |
| REPARAMETRISED_COMPOSITE_CURVE_SEGMENT | RCCS |
| REPLICATE_FEATURE | RPLFTR |
| REPRESENTATION | RPRSNT |
| REPRESENTATION_CONTEXT | RPRCNT |
| REPRESENTATION_ITEM | RPRITM |
| REPRESENTATION_MAP | RPRMP |
| REQUIREMENT_FOR_ACTION_RESOURCE | RFAR |
| 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_TOP | RBTP |
| RIB_TOP_FLOOR | RBTPFL |
| ROLE_ASSOCIATION | RLASS |
| ROUND_HOLE | RNDHL |
| ROUNDED_END | RNDEND |
| ROUNDED_U_PROFILE | RNUPR |
| ROUNDNESS_TOLERANCE | RNDTLR |
| RUNOUT_ZONE_DEFINITION | RNZNDF |
| RUNOUT_ZONE_ORIENTATION | RNZNOR |
| RUNOUT_ZONE_ORIENTATION_REFERENCE_DIRECTION | RZORD |

 

## Table 2 — AIM short names (continued)

| | |
|---|---|
| SEAM_CURVE | SMCRV |
| SECURITY_CLASSIFICATION | SCRCLS |
| SECURITY_CLASSIFICATION_ASSIGNMENT | SCCLAS |
| SECURITY_CLASSIFICATION_LEVEL | SCCLLV |
| SEQUENTIAL_METHOD | SQNMTH |
| SERIAL_ACTION_METHOD | SRACMT |
| SHAPE_ASPECT | SHPASP |
| SHAPE_ASPECT_DERIVING_RELATIONSHIP | SADR |
| SHAPE_ASPECT_RELATIONSHIP | SHASRL |
| SHAPE_DEFINING_RELATIONSHIP | SHDFRL |
| SHAPE_DEFINITION_REPRESENTATION | SHDFRP |
| SHAPE_DIMENSION_REPRESENTATION | SHDMRP |
| SHAPE_REPRESENTATION | SHPRPR |
| SHAPE_REPRESENTATION_WITH_PARAMETERS | SRWP |
| SHELL_BASED_SURFACE_MODEL | SBSM |
| SHELL_BASED_WIREFRAME_MODEL | SBWM |
| SHELL_BASED_WIREFRAME_SHAPE_REPRESENTATION | SBWSR |
| SI_UNIT | SUNT |
| SINGLE_ACTIVITY_RELATIONSHIP | SNACRL |
| SLOT | SLOT |
| SLOT_END | SLTEND |
| SOLID_ANGLE_UNIT | SLANUN |
| SOLID_MODEL | SLDMDL |
| SPHERICAL_CAP | SPHCP |
| SPHERICAL_SURFACE | SPHSRF |
| SQUARE_U_PROFILE | SQUPR |
| STANDARD_UNCERTAINTY | STNUNC |
| STEP | STEP |

## Table 2 — AIM short names (continued)

| | |
|---|---|
| STRAIGHTNESS_TOLERANCE | STRTLR |
| SURFACE | SRFC |
| SURFACE_CURVE | SRFCRV |
| SURFACE_OF_LINEAR_EXTRUSION | SL |
| SURFACE_OF_REVOLUTION | SROFRV |
| SURFACE_PATCH | SRFPTC |
| SURFACE_PROFILE_TOLERANCE | SRPRTL |
| SURFACE_REPLICA | SRFRPL |
| SWEPT_SURFACE | SWPSRF |
| SYMMETRIC_SHAPE_ASPECT | SYSHAS |
| SYMMETRY_TOLERANCE | SYMTLR |
| TANGENT | TNGNT |
| TAPER | TAPER |
| TEE_PROFILE | TPRF |
| THREAD | THREAD |
| TIME_MEASURE_WITH_UNIT | TMWU |
| TIME_UNIT | TMUNT |
| TOLERANCE_VALUE | TLRVL |
| TOLERANCE_ZONE | TLRZN |
| TOLERANCE_ZONE_DEFINITION | TLZNDF |
| TOLERANCE_ZONE_FORM | TLZNFR |
| TOOL_ASSEMBLY | TLASS |
| TOOL_ASSEMBLY_ELEMENT | TLASEL |
| TOOL_ASSEMBLY_RELATIONSHIP | TLASRL |
| TOOL_BODY | TLBDY |
| TOOL_CONTRACT | TLCNT |
| TOOL_MAGAZINE_TURRET_CAROUSEL | TMTC |
| TOOL_PLACEMENT_INSTRUCTION | TLPLIN |

 

## Table 2 — AIM short names (concluded)

| | |
|---|---|
| TOOL_SETUP | TLSTP |
| TOPOLOGICAL_REPRESENTATION_ITEM | TPRPIT |
| TOROIDAL_SURFACE | TRDSRF |
| TOTAL_RUNOUT_TOLERANCE | TTRNTL |
| TRANSITION_FEATURE | TRNFTR |
| TRIMMED_CURVE | TRMCRV |
| TURNED_KNURL | TRNKNR |
| TYPE_QUALIFIER | TYPQLF |
| UNCERTAINTY_MEASURE_WITH_UNIT | UMWU |
| UNCERTAINTY_QUALIFIER | UNCQLF |
| UNIFORM_CURVE | UNFCRV |
| UNIFORM_SURFACE | UNFSRF |
| VALIDATION | VLDTN |
| VALUE_RANGE | VLRNG |
| VALUE_REPRESENTATION_ITEM | VLRPIT |
| VECTOR | VECTOR |
| VEE_PROFILE | VPRF |
| VERSIONED_ACTION_REQUEST | VRACRQ |
| VERTEX | VERTEX |
| VERTEX_LOOP | VRTLP |
| VERTEX_POINT | VRTPNT |
| VERTEX_SHELL | VRTSHL |
| VIEW_REFERENCE | VWRFR |
| WIRE_SHELL | WRSHL |
| WORK_CELL | WRKCLL |
| WORKSTATION | WRKSTT |

# Annex C
(normative)

# Implementation method specific requirements

The implementation method defines what types of exchange behaviour are 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 in the AIM 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 'process_planning_schema'.

 

# Annex D
(normative)

# Protocol Implementation Conformance Statement (PICS) 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:

1. For simplicity of reference, an identifier for the product or system with which the tested STEP implementation is packaged in and/or procured by is required.

Product/system identifier (or name): _____

2. There are ten 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: Class 6 and shapes represented by advanced b-rep;

Class 2: Class 6 and shapes represented by non topological surface and wireframe models;

Class 3: Class 6 and shapes represented by wireframe models with topology;

Class 4: Class 6 and shapes represented by manifold surface models with topology;

Class 5: Class 6 and shapes represented by faceted b-rep;

Class 6: NC process plan information without shape.

3. Conformance to this international standard may be realized in one or more of several different implementation methods. The implementation methods define what types of exchange behavior are required with respect to this international standard.

Claimed implementation forms - circle choices:

exchange structure (ISO 10303-21);

exchange structure (ISO 10303-22);

exchange structure (ISO 10303-28).

4. If the exchange structure used is ISO 10303-28, which one?

ISO 10303-28 exchange structure? : _____

5. 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: _____

6. 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

7. If yes, which ones?

(a)Conformance class(es): _____

(b) Default Response: _____

8. A statement of conformance shall include identification of at least one party deeming conformance for the implementation.

Evaluator(s) (tester/certifier/accrediter):_____

 

# 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(240) version(0)}

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

## E.2 Schema identification

To provide for unambiguous identification of the process_planning_schema in an open system, the object identifier

{iso standard 10303 part(240) version(0) object(1) process-planning-schema(1)}

is assigned to process_planning_schema expanded schema (see Annex A).

{iso standard 10303 part(240) version(0) object(1) process-planning-schema(2)}

is assigned to process_planning_schema short form schema (see clause 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 as an aid to understanding the scope and information requirements defined in this application protocol. The model is presented as a set of figures that contain the activity diagrams and a set of definitions of the activities and their data. Activities and data flows that are out of scope are marked with an asterisk.

The AAM covers activities which go beyond the subject of this application protocol. The diagrams use a modified IDEF0 notation [11]. 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 application protocol.

This is an activity model of life cycle activities across all process planning. 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.

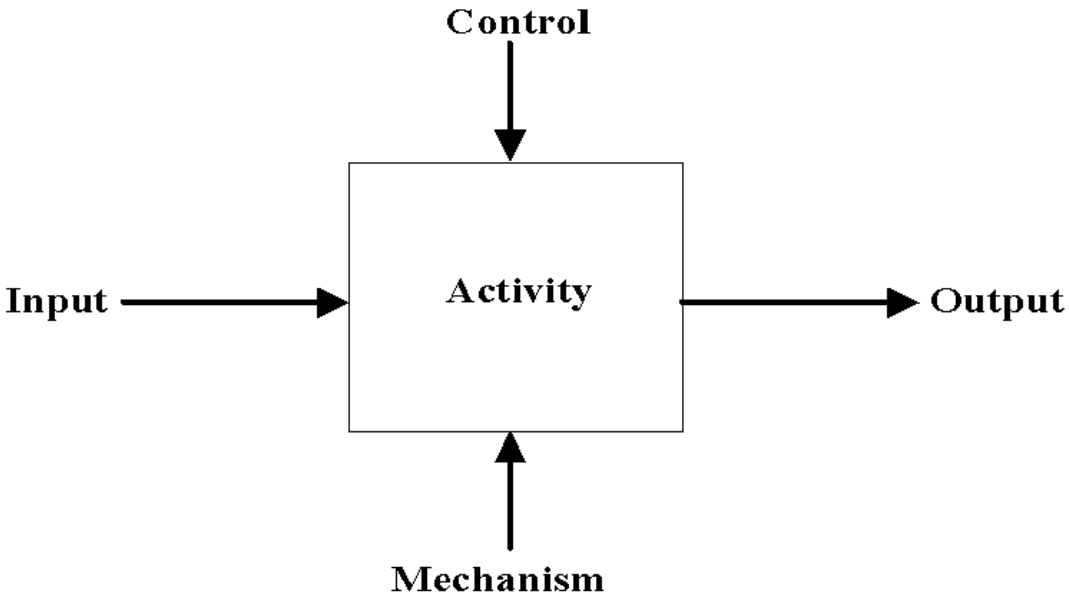


**Figure F.1 — IDEF0 Basic notation**

 

## F.1 Application activity model definitions and abbreviations

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions given in this Annex do not supersede the definitions given in the main body of the text.

**F.1.1 Assign Features to Process (A1153)**: associate the processes with manufacturing features,the related datum and tolerances to be manufactured in each process, and tolerances

**F.1.2 CAPP/CAM***: Computer Aided Process Planning / Computer Aided Manufacturing

**F.1.3 Capture Manufacturing Features (A111)**: recast the part geometry and topology in terms of features to be created and /or zones of material to be removed from stock materials.

**F.1.4 Controllers***: controllers for automated machining, assembly, handling and inspection systems, such as machine tools, wire layers, welders, robots, and coordinate measuring machines

**F.1.5 Control Programs**: programs for the direct control of automated machining, assembly, handling and inspection systems, such as machine tools, wire layers, welders, robots, and coordinate measuring machines

**F.1.6 Cost/Facility Reports***: reports on the production cost and the utilization of facilities

**F.1.7 Customer Order Status***: progress status of the customer order

**F.1.8 Data Model Schemas**: data structure for modeling product, process, resources, etc

**F.1.9 Data Model Schemas for Assembly or Fabrication Process**: data models to describe process data for assembly or fabrication process

**F.1.10 Data Model Schemas for Conditioning or Finishing Process**: data models to describe process data for conditioning or finishing process

**F.1.11 Data Model Schemas for Forming Process**: data models to describe process data for forming process

**F.1.12 Data Model Schemas for Inspection**: data models to describe process data for inspection

**F.1.13 Data Model Schemas for Material Removal**: data models to describe process data for material removal

**F.1.14 Data Model Schema for Manufacturing Feature***: data model to describe defined shapes for material removal.

**F.1.15 Data Model Schemas for Manufacturing Process**: data models to describe process data for manufacturing methods and inspection

**F.1.16 Data Model Schemas for Welding Process**: data models to describe process data for welding process

 

**F.1.17 Define In-Process Workpiece Configurations (A151)**: define intermediate geometries of a workpiece at some stages in a process and associated dimensions, tolerances, and surface finishes which are important to later processes.

**F.1.18 Design Special Tooling (A143)**: where the process plans require tooling that cannot be acquired off-the-shell or made up from standard components, specify the functional and physical requirements of this special tooling and initiate the corresponding design and engineering processes.

**F.1.19 Design Tooling Assemblies (A142)**: where the process plans require tooling made-up from standard components, generate the tool or fixture assembly drawings, component specifications and setting specifications.

**F.1.20 Determine Detail Data for Tasks (A12132)**: specify detail operation data (the parameters for task) for each task.

**F.1.21 Develop Equipment Instructions (A15)**: for each station in the routing plan, define and document exactly what steps must be taken by the machines and by the human operators and specialists, in order to accomplish the operation sequence assigned to that station.

**F.1.22 Develop Final Cost Estimation (A161)**: determine the manufacturing cost of the part in terms of materials, time and resources. Based on the time estimates for use of manufacturing resources and skilled labor, plus tooling, materials, handling and in-work transportation costs, estimate the total cost of production of the part.

**F.1.23 Develop Machine Setup (A1154)**: specify activities for machine setup to manufacture the product in each process.

**F.1.24 Develop Preliminary Cost Estimation (A116)**: estimate the processing time, materials, equipment and manpower costs for major manufacturing methods selected. This may result in recommendation for design changes to permit less costly processes to be used.

**F.1.25 Develop Strategy for Process Plan (A113)**: develop the strategy for the overview of the process plan, that is, strategies for selection and sequence of manufacturing methods, selection of resources, design of clamping positions, feature assignments, etc

**F.1.26 Develop Strategy for Tooling Selection (A1155)**: develop the strategy for the identification of individual tools, jigs, fixtures and gages required in the manufacturing of a product (through the specification of a recognizable reference number).

**F.1.27 Develop Tooling Packages (A14)**: complete specifications for the required tooling, including cutting tools, molds, dies, fixtures, end-effectors, probes, sensors, etc.

**F.1.28 Engineer Assembly or Fabrication Process (A1214)**: specify the component operations of the assembly or the fabrication process, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence of the operations.

**F.1.29 Engineer Conditioning or Finishing Process (A1215)**: specify the component operations of the conditioning process o9r the finishing process, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence of the operations.

 

**F.1.30 Engineer Forming Process (A1211)**: specify the component operations of the forming process, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence of the operations.

**F.1.31 Engineer Inspection Process (A122)**: for the inspection process, specify the component operations, including the associated manufacturing features and/or parameters which characterize the operations.

**F.1.32 Engineer Manufacturing Methods & Part Routing (A11)**: define the major processes involved in making the part, and identify the types of machines and special skills to be used. Identify stock materials or components to be used, select the major processes to be performed, and define the major part routing plan, the sequence of workstations or machines that the part must go through.

**F.1.33 Engineer Manufacturing Process (A1)**: define the process of making the product, including the elementary stock materials and components to be acquired, the equipment, tooling and skills to be used and the details of that usage. Details include the exact sequence of setups and operations to be performed, and the complete instructions for each operation, whether by human or automated resources.

**F.1.34 Engineer Material Removal (A1213)**: specify the component operations of the material removal process, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence of the operations.

**F.1.35 Engineer New Process (A13)***: determine how to perform a new process by designing new or modified machines, new tools and end-effectors, new measurements and process controls.

**F.1.36 Engineer Process in Detail (A12)**: define and validate the sequence of operations that makeup each major process, as defined by the process selection and equipment selection.

**F.1.37 Engineer Production System (A2)***: design new or modified facilities for the manufacture of a particular collection of parts. A facility may be a plant, a shop, a line, a manufacturing cell, or a group of manufacturing cells.

**F.1.38 Engineer Welding Process (A1212)**: specify the component operations of that method, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence of the operations.

**F.1.39 Engineering Product Definition Data (PDD)**: the subset of released Engineering Product Definition Data (PDD) relevant to Process Planning and a superset of change descriptions. It includes Engineering Bill of Material (EBOM), shape, interface requirements, change description, geometry model, tolerances, surface finish, material requirements, and specified stock material.

**F.1.40 Engineering PDD Change Request** : a formal form used to document and control requested changes to the product definition data Part of Engineering PDD.

**F.1.41 Equipment Availability**: a report containing the status of available equipments at particular point in time.

**F.1.42 Equipment Change Request**: a formal method of documenting and controlling requested changes to the equipment

**F.1.43 Equipment Selections**: the selected equipments (machines or workstations) for each process to required to manufacture the product

**F.1.44 ERP/MRP2***: a enterprise resource planning system/ a manufacturing resource planning system

**F.1.45 Estimate Machining Time (A12134)**: estimate machining time of the material removal process from the tasks and detail data for tasks.

**F.1.46 Estimate Tooling Cost (A144)**: estimate the cost of the tooling required for some volume of part production. For off-the-shell tooling and assemblies, this includes materials, preparation, storage and handling, and decommissioning. For special tooling, it also includes the cost of design, engineering and production of the tooling.

**F.1.47 Facility Cost Estimates***: estimated cost to manufacture or procure facilities required to manufacture the product

**F.1.48 Facility Design***: the design of facilities as an output from engineering production system required to manufacture the product

**F.1.49 Facility ID**: a unique identification of a single facility

**F.1.50 Facility Implementation Plan***: the schedule to implement facilities into the factory.

**F.1.51 Facility Orders***: notification to procure facilities required to manufacture the product

**F.1.52 Facility/Tooling***: a specific manufacturing site or sites, machines or workstations Cutting tools, jigs, fixtures and gages required in the manufacturing of a product

**F.1.53 Finalize Manufacturing Data Package (A16)**: produce accurate time and cost estimates, and assemble the engineering information into the required packages for production use. Each package is reviewed and signed off.

**F.1.54 Finalize Part Routing (A123)**: merge the manufacturing processes and the inspection process, and specify the overall sequence of all operations.

**F.1.55 Generate Machine Programs (A154)**: generate programs for the direct control of automated machining, assembly, handling and inspection systems, such as machine tools, wire layers, welders, robots, and coordinate measuring machines.

**F.1.56 Generate Operator Instructions (A153)**: specify the sequence of fixturing and processing steps to be executed at each workstation. Create operation sheets for shop-floor operators.

**F.1.57 Generate Tasks for Material Removal (A12131)**: generate tasks for material removal for each feature assigned to the process.

**F.1.58 Inspection Data**: data that specifies the component operations for the inspection, including the associated manufacturing features and/or parameters which characterize the operations

**F.1.59 Initial Materials***: a definition of the initial stock as called out in the process plan.

 

**F.1.60 Instruction Change Request**: a formal method of documenting and controlling requested changes to the instruction

**F.1.61 Instruction Validation Results**: validation information for the operator and equipment instructions generated from the validation of equipment instructions

**F.1.62 Manufacture Mechanical Products (A0)**: defines and processes all data and interfaces necessary for manufacturing mechanical products.

**F.1.63 Manufactured Product***: the commodity produced as a result of one or more manufacturing operations being applied to a piece of raw material or previously produced components

**F.1.64 Manufacturing Data Package Requirements**: requirements for final cost estimates, resource planning package, scheduling package and plan library

**F.1.65 Manufacturing Features**: identifies a volume of material that must be removed to obtain the final geometry from the initial stock.

**F.1.66 Manufacturing Methods Change Request**: a formal method of documenting and controlling requested changes to the manufacturing methods

**F.1.67 Manufacturing Process Features**: manufacturing features which have been assigned to a process

**F.1.68 Manufacturing Resource Database**: a database to store the specifications of machines or workstations, cutting tools, jigs, fixtures, etc.

**F.1.69 Manufacturing Technology Database (Manufacturing Knowledge)**: a database to store the manufacturing methods for each manufacturing feature to be referred from CAPP/CAM

**F.1.70 Material Selections**: the selected material for the part required to manufacture the product

**F.1.71 Material/Tooling Inventory**: the information to obtain necessary toolings and materials needed to support manufacturing

**F.1.72 Material/Tooling orders***: notification to procure material and tooling required to manufacture the product

**F.1.73 MES***: a manufacturing execution system

**F.1.74 New Process Model***: a process model that defines a new process engineered.

**F.1.75 New Process Requirements***: needs for items to engineer new processes

**F.1.76 Operation Change Request**: a formal method of documenting and controlling requested changes to the operation

**F.1.77 Operate Production System (A5)***: based on the production plan and master schedule carry out manufacturing orders in a production system to produce finished goods. This activity includes initiate, manage, and report on production activities.

**F.1.78 Operation Sheets**: the documents for operators containing all the data sets that define the required tooling, fixtures, setups, operations, and machine control parameters

**F.1.79 Operators***: operators working on the shop floor

**F.1.80 Plan Library**: a complete data package of the process plan for a product

**F.1.81 Plan Production (A3)***: analyze parts and perform make/buy decisions for all the parts. Develop business plan and schedule to acquire necessary resources and/or to produce products for the market. Resources include material, finished parts, equipment, and labor skills.

**F.1.82 Planning Policies***: the policies for planning the production of a product (ex. make/buy decision of parts)

**F.1.83 Preliminary Cost Estimates**: the production cost that is estimated preliminary. This may result in recommendation for design changes to permit less costly processes to be used.

**F.1.84 Process Change Request**: a formal method of manufacturing process requested changes to the tooling. It is composed of methods change request and operation change request.

**F.1.85 Process Data for Operation**: the data describing the operations and their sequence for manufacture of features that are assigned to each manufacturing process.

**F.1.86 Process Data for Part Routing**: the data describing the selected manufacturing methods (manufacturing processes) and their sequence (part routing), and the information of each manufacturing process containing the selected machine or workstation, clamping positions, jigs and fixtures, assigned manufacturing features, etc.

**F.1.87 Procure Product Material & Facility (A4)***: based on resource requirement plan, procure product materials and facilities from suppliers to meet the production schedule.

**F.1.88 Procured Facility/Tooling**: the facility and tooling that are procured by facility orders and tooling orders.

**F.1.89 Procured Initial Material***: initial materials that are procured by the material/tooling orders

**F.1.90 Production BOM**: a definition of all materials, parts, components and special tools necessary to manufacture a given product.

**F.1.91 Production Cost Estimates** : the estimated total cost to manufacture a product

**F.1.92 Production Orders**: the initial documentation or information necessary to start a project requiring the manufacture of a piece part. A request from either an enterprise or external person or organization requesting manufacturer and /or design of a particular product.

**F.1.93 Production Requirements***: needs for items to engineer a production system

**F.1.94 Production Schedule***: the timetables for operating a production system

**F.1.95 Production Time Estimates**: the estimated time to manufacture the product

 

**F.1.96 Release Resource Package (A162)**: summarize resource requirements: equipment and manpower time, tooling and materials quantities, expected yields, etc. to provide inputs for each planning process.

**F.1.97 Release Scheduling Package (A163)**: summarize production workflow requirements: equipment routing, together with machine usage, manpower skills and time, tooling, materials, etc. for each station in the routing.

**F.1.98 Resource Requirements**: needs for items to manufacture a product: equipment and manpower time, tooling and materials quantities, expected yields, etc. to provide inputs for each planning processes

**F.1.99 Scheduling Package**: a collection of all the necessary data required for a production schedule. It contains equipment routing, together with machine usage, manpower skills and time, tooling, materials, etc. for each station in the routing

**F.1.100 SCM***: a supply chain management system

**F.1.101 Select Equipment (A1152)**: the identification or individual items of equipment required to manufacture the product. This selection process is constrained by the previously selected manufacturing facility (and by the current properties or characteristics).

**F.1.102 Select Facility (A1151)**: the identification of an enterprise's specific manufacturing site or sites that will be used in the manufacture of the subject product. This selection will be made based on consideration of business or economic factors and site-specific or unique manufacturing capabilities.

**F.1.103 Select and Sequence Manufacturing Methods (A114)**: choose the manufacturing methods that transform material stocks into final product, and sequence them.

**F.1.104 Select Resources (A115)**: the identification of goods and services required to manufacture the product. This function provides facilities to house the production process and support services, and equipments to transform the material into products; cutting tools, jigs, fixtures, and gages.

**F.1.105 Select Stock Materials (A112)**: identify the specific stock materials and off the shelf components to be used and their quantities.

**F.1.106 Select Tooling Assemblies (A141)**: identify off-the-shell tooling assemblies which will meet the requirements of the process plans, as opposed to tooling which must be designed specially for the purpose.

**F.1.107 Simulation System***: a simulation system for engineering a production system

**F.1.108 Special Tooling Designs**: the design of special toolings as an output from design of special tooling required to manufacture the product

**F.1.109 Specify and Sequence Operations (A121)**: for each manufacturing methods selected except inspection process, specify the component operations of that method, including the associated manufacturing features and/or parameters which characterize the operations, and then specify the sequence in which the operations will be performed. Special tooling requirements may be identified in this activity.

**F.1.110 Specify Tooling Requirements (A152)**: specify the tooling, end-effectors, fixtures and consumable materials required at the processing station.

**F.1.111 Sequence Tasks (A12133)**: specify the sequence of all tasks for the process.

**F.1.112 Strategy for Process Plan**: the strategy for the overview of the process plan, that is, strategies for selection and sequence of manufacturing methods, selection of resources, design of clamping positions, feature assignments, etc.

**F.1.113 Strategy for Tooling Selections**: the strategy for the selection of toolings.

**F.1.114 Time & Cost Constraints**: the collective information on time and cost related limitations, that are necessary in the production planning process

**F.1.115 Tooling Assembly Designs**: the design of tooling assembly as an output from design of tooling assemblies required to manufacture the product

**F.1.116 Tooling Change Request**: a formal method of documenting and controlling requested changes to the tooling

**F.1.117 Tooling Cost Estimates**: estimated cost to manufacture and/or procure toolings required to manufacture the product

**F.1.118 Tooling Designs**: the design of toolings as an output from design of tooling assemblies and special tooling required to manufacture the product

**F.1.119 Tooling Requirements**: the criteria for standard tooling, perishable tooling, designed tooling, or special tooling, which are necessary for each setup

**F.1.120 Tooling Selections**: the selected tooling for the part required to manufacture the product

**F.1.121 Update Plan Library (A164)**: complete plan package and enter the new approved plans into the local plan reference library or archive.

**F.1.122 Validate and Approve Process Plan (A124)**: review and evaluate the process plan against part quality and performance objectives. This activity includes the identification of desirable modifications and/or the approval of the process plans.

**F.1.123 Validate Equipment Instructions (A155)**: verify and approve for production use the operator and equipment instructions and all the data sets that define the required tooling, fixtures, setups, operations, and machine control parameters.

**F.1.124 Validate Run Requirements**: product validation requirements that manufacturing process must follow or adhere

**F.1.125 Validation Run Results**: product validation information generated from the actual manufacturing execution of process plan operations

**F.1.126 Workpiece Configuration**: intermediate geometries of a workpiece at some stages in a process and associated dimensions, tolerances, and surface finishes which are important to later processes

## F.2 Application activity model diagrams

 

The application activity model diagrams are given in figures F.2 through F.12. The graphical form of the Application Activity Model is presented in the IDEF0 activity modelling format. Activities and data flows that are out of scope are marked with asterisks.

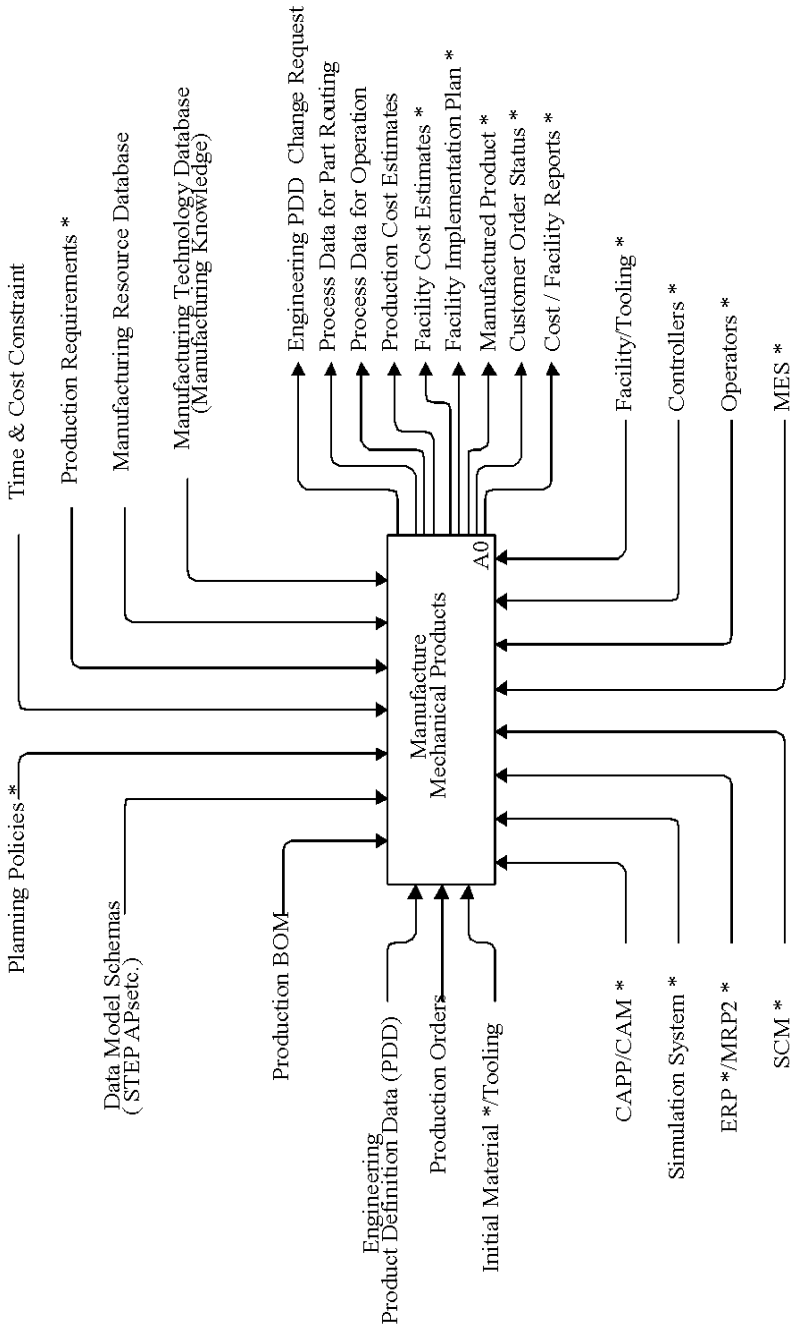


**Figure F.2 — A-0 Process plans for machined parts**

 

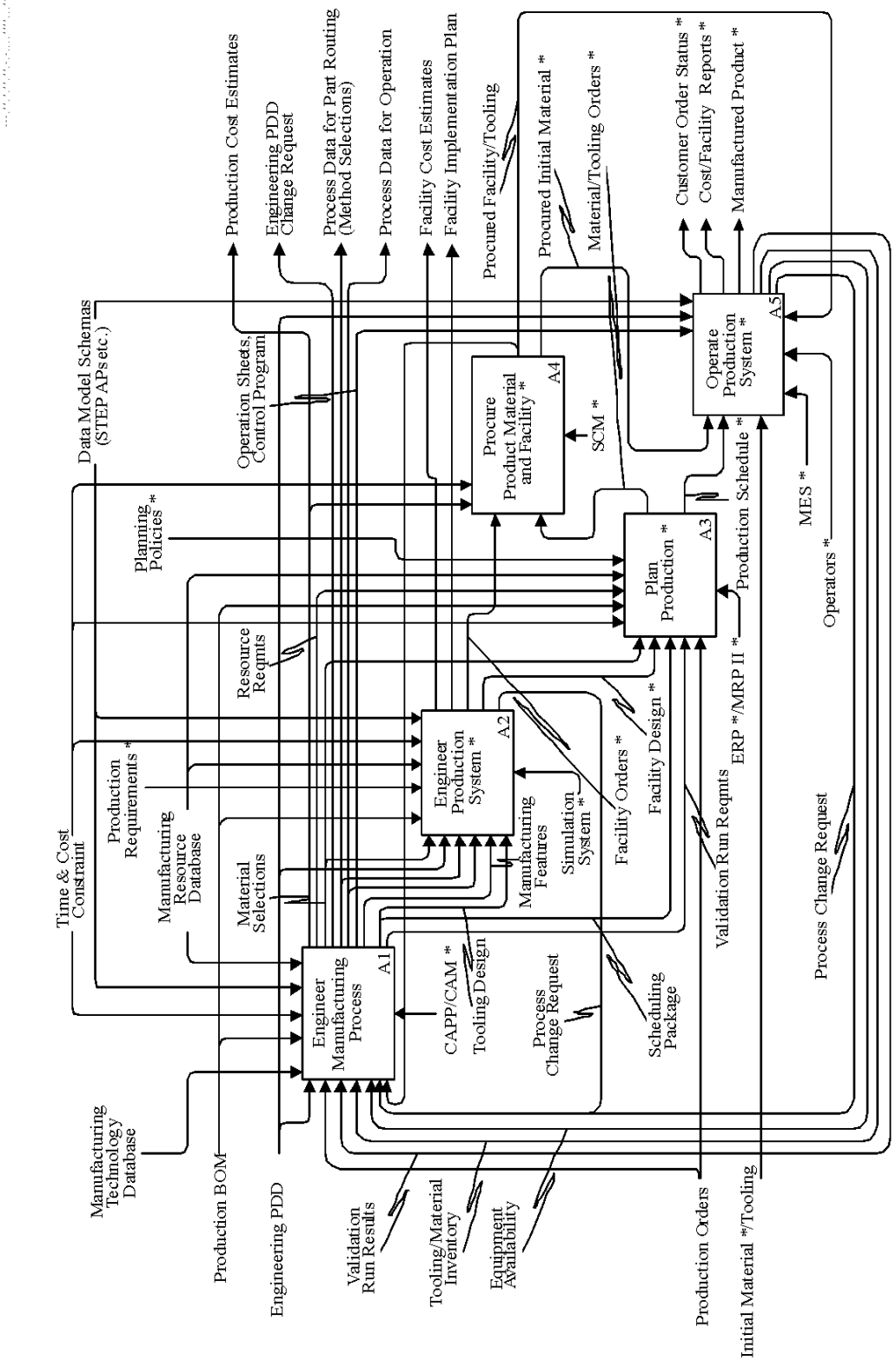


**Figure F.3 — A0 Manufacture Mechanical Products**

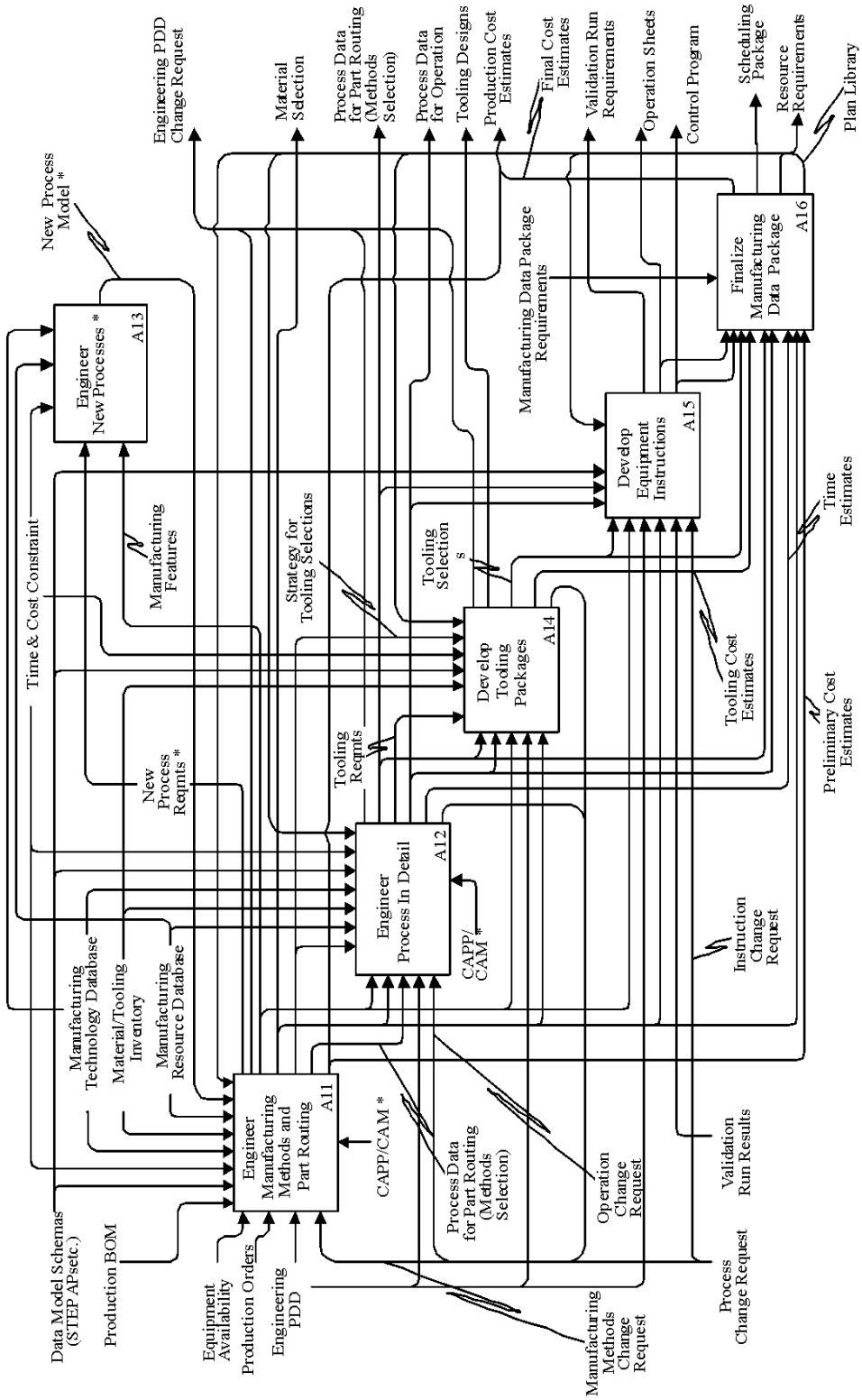


**Figure F.4 — A1 Engineer Manufacturing Process**

 

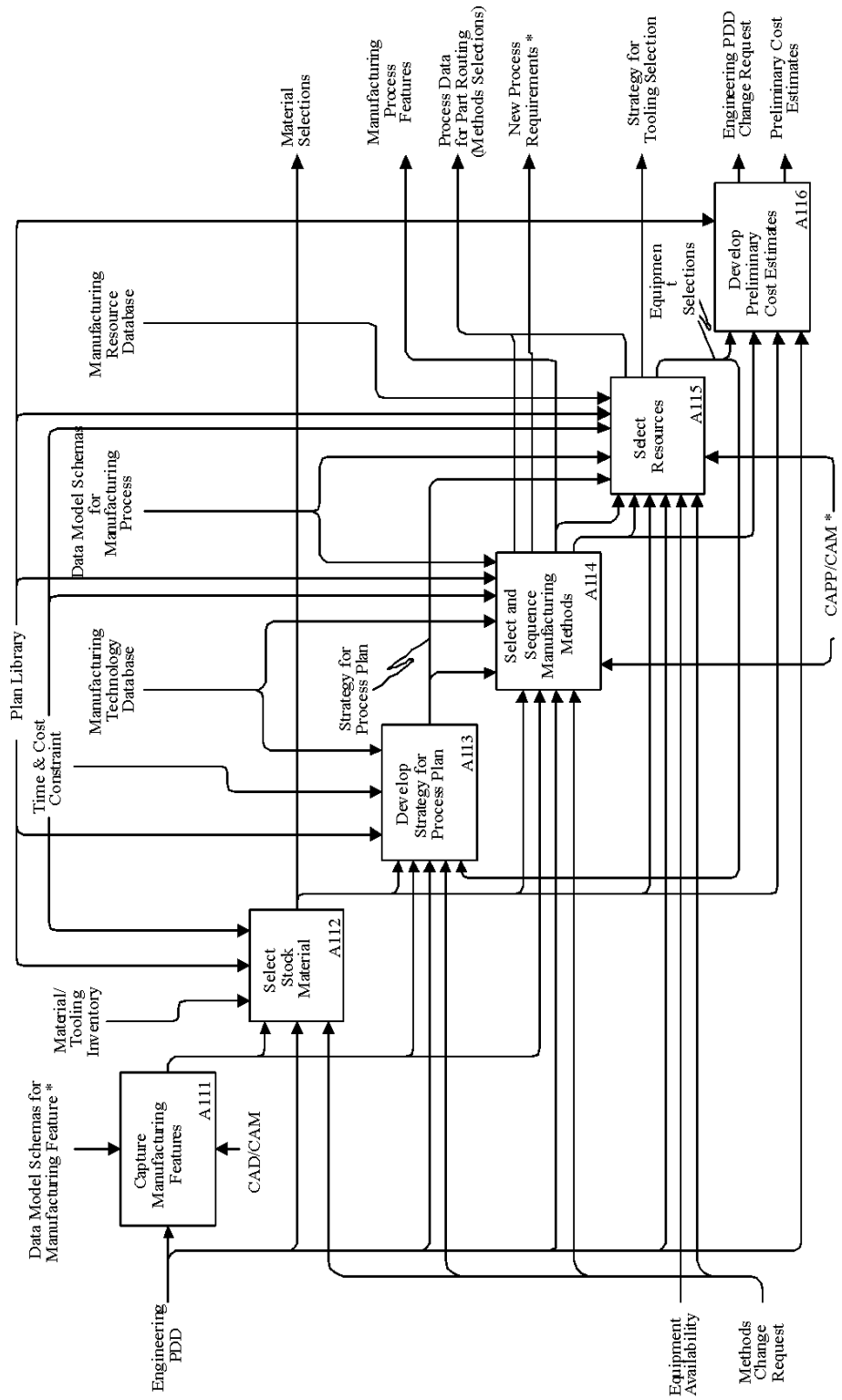


**Figure F.5 — A11 Engineer Manufacturing Methods and Part Routing**

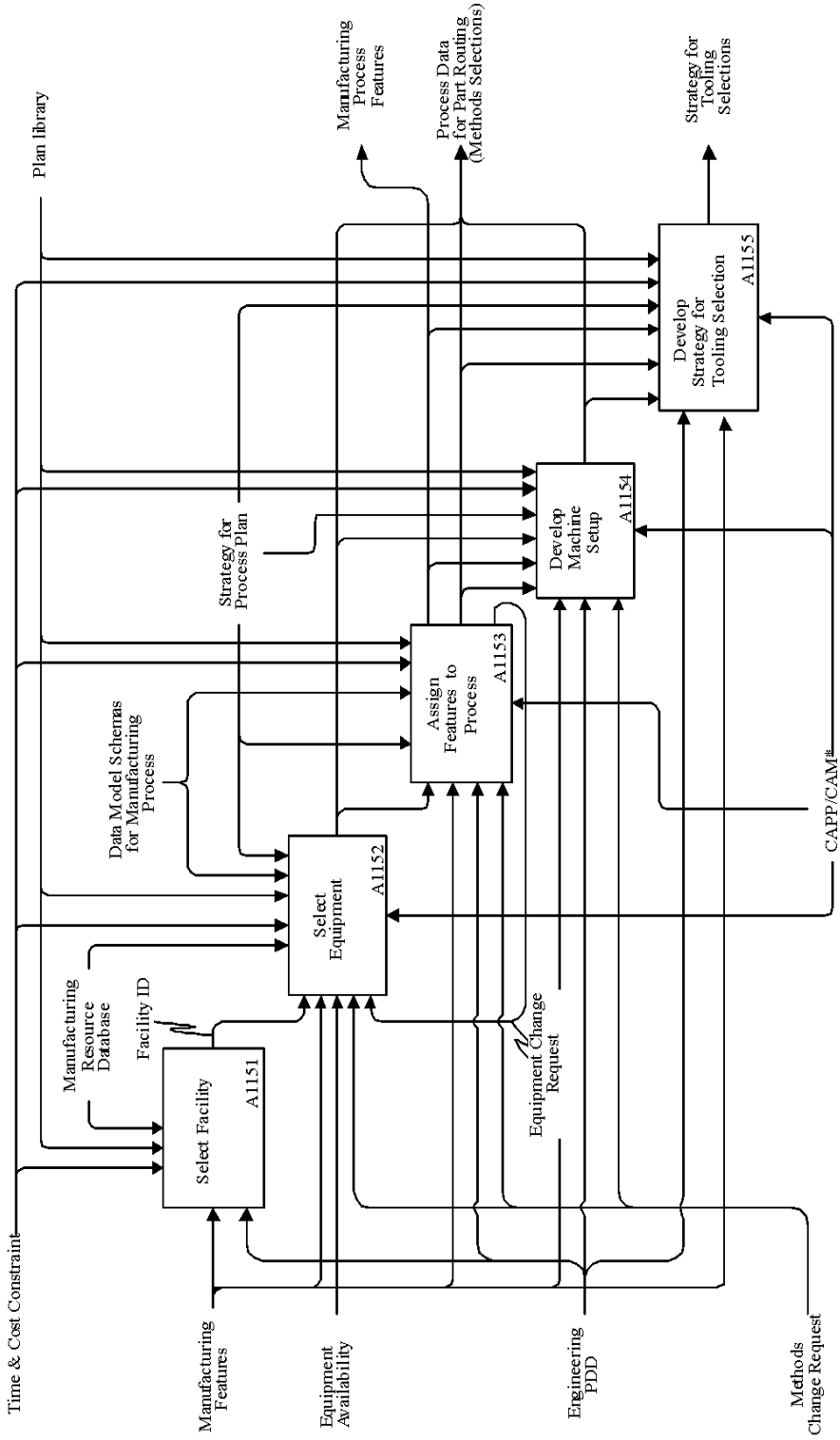


**Figure F.6 — A115 Select Resources**

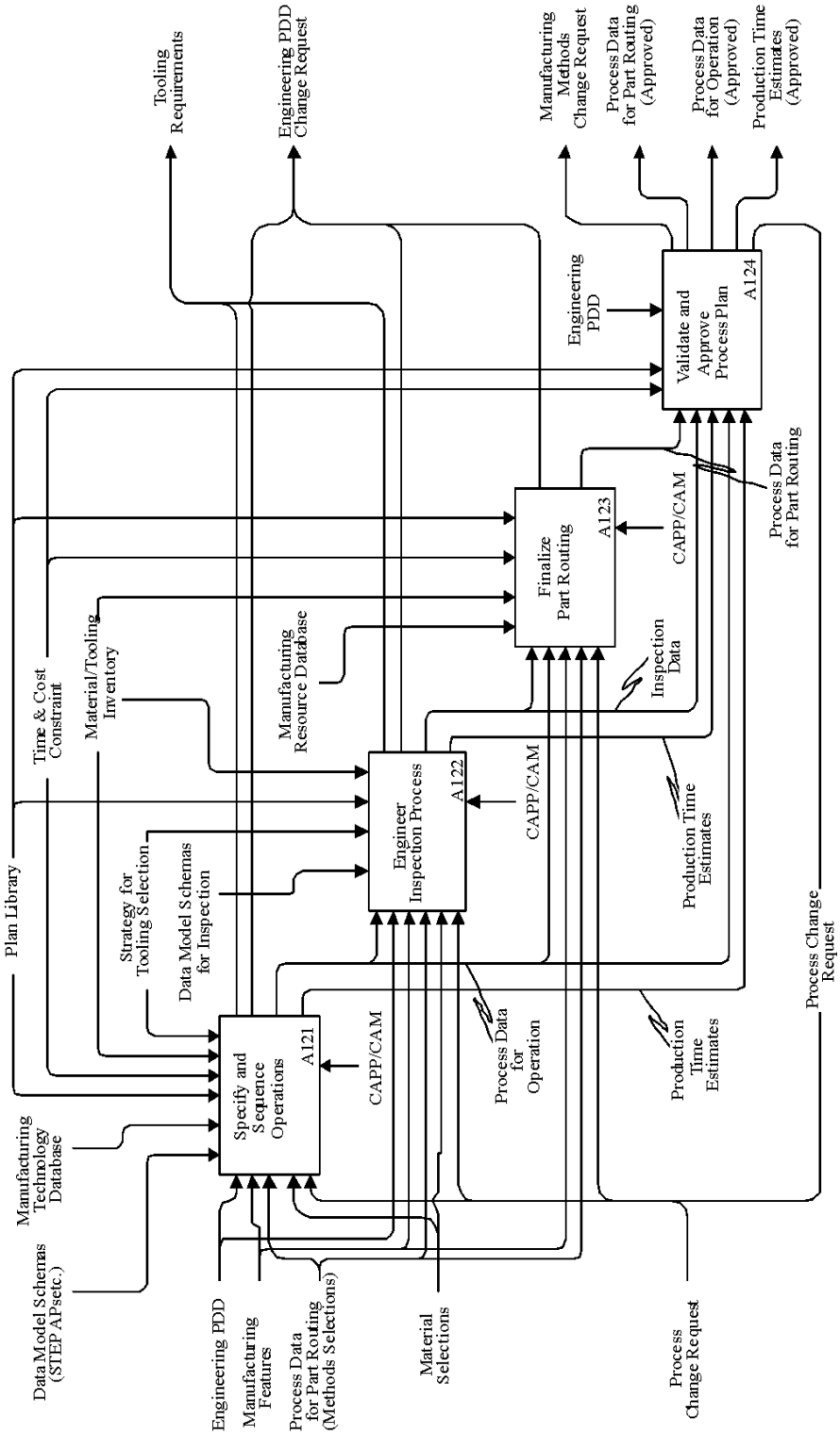


**Figure F.7 — A12 Engineer Process Detail**

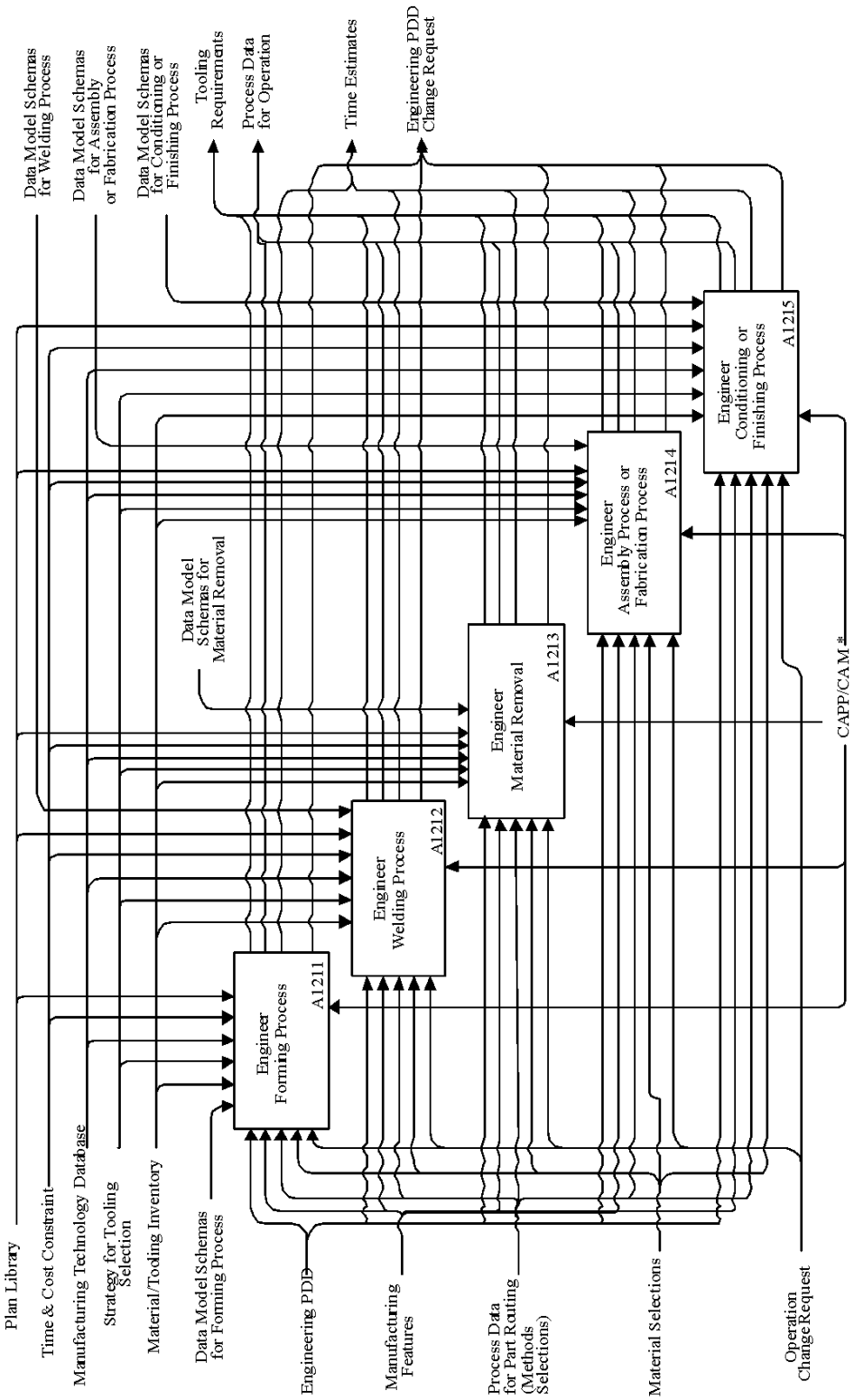


**Figure F.8 — A121 Specify and Sequence Operations**

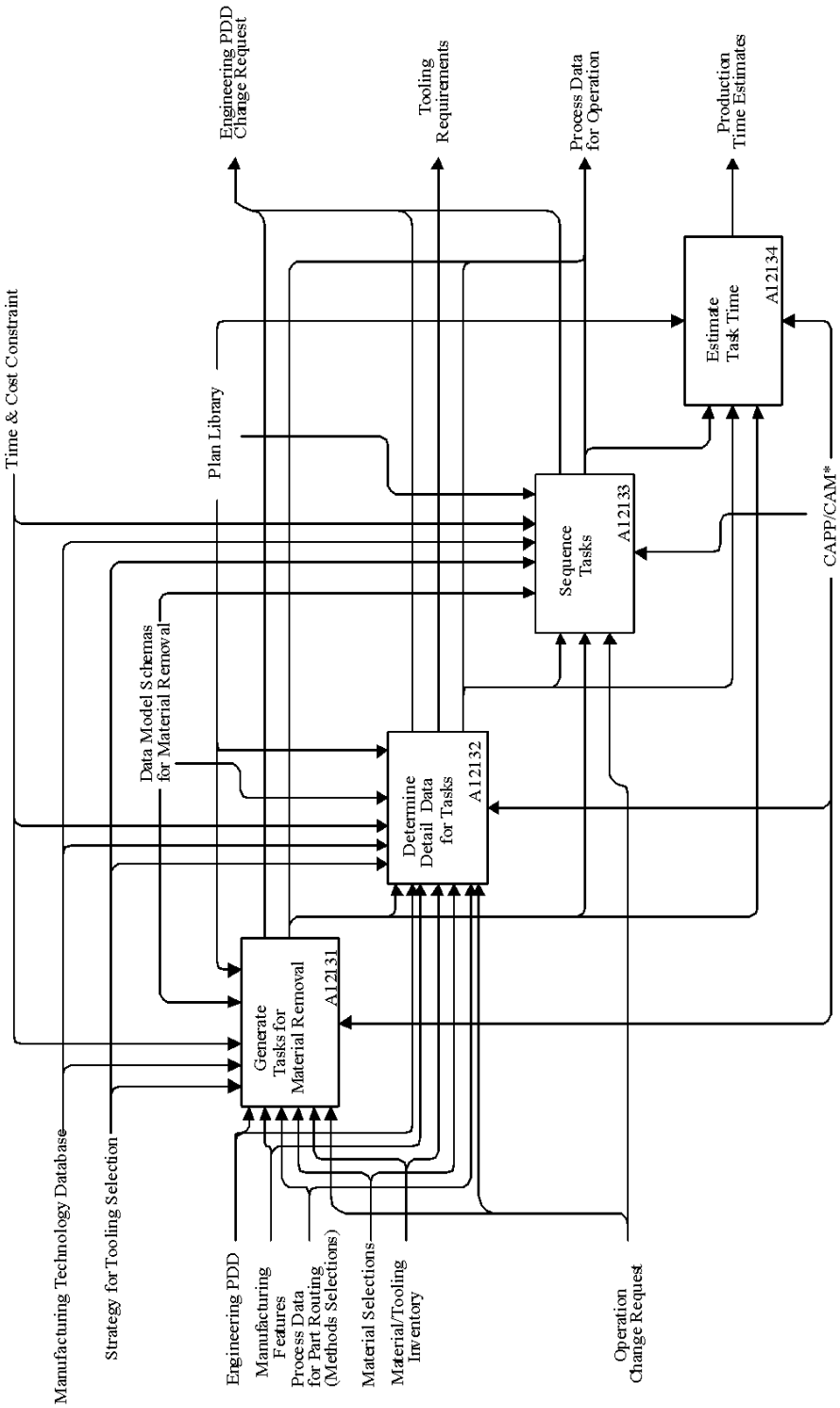


**Figure F.9 — A1213 Engineer Material Removal**

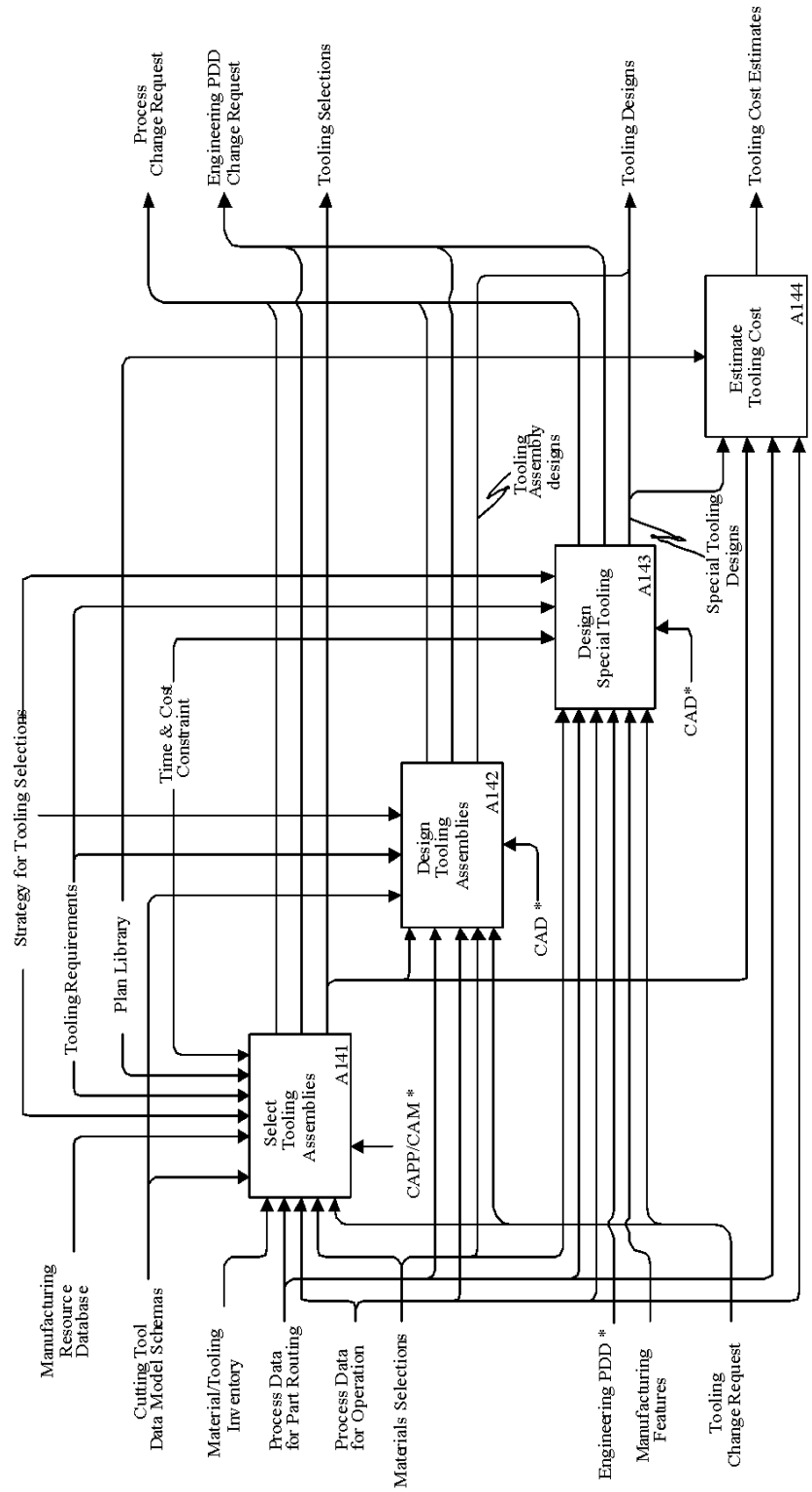


**Figure F.10 — A14 Develop Tooling Packages**

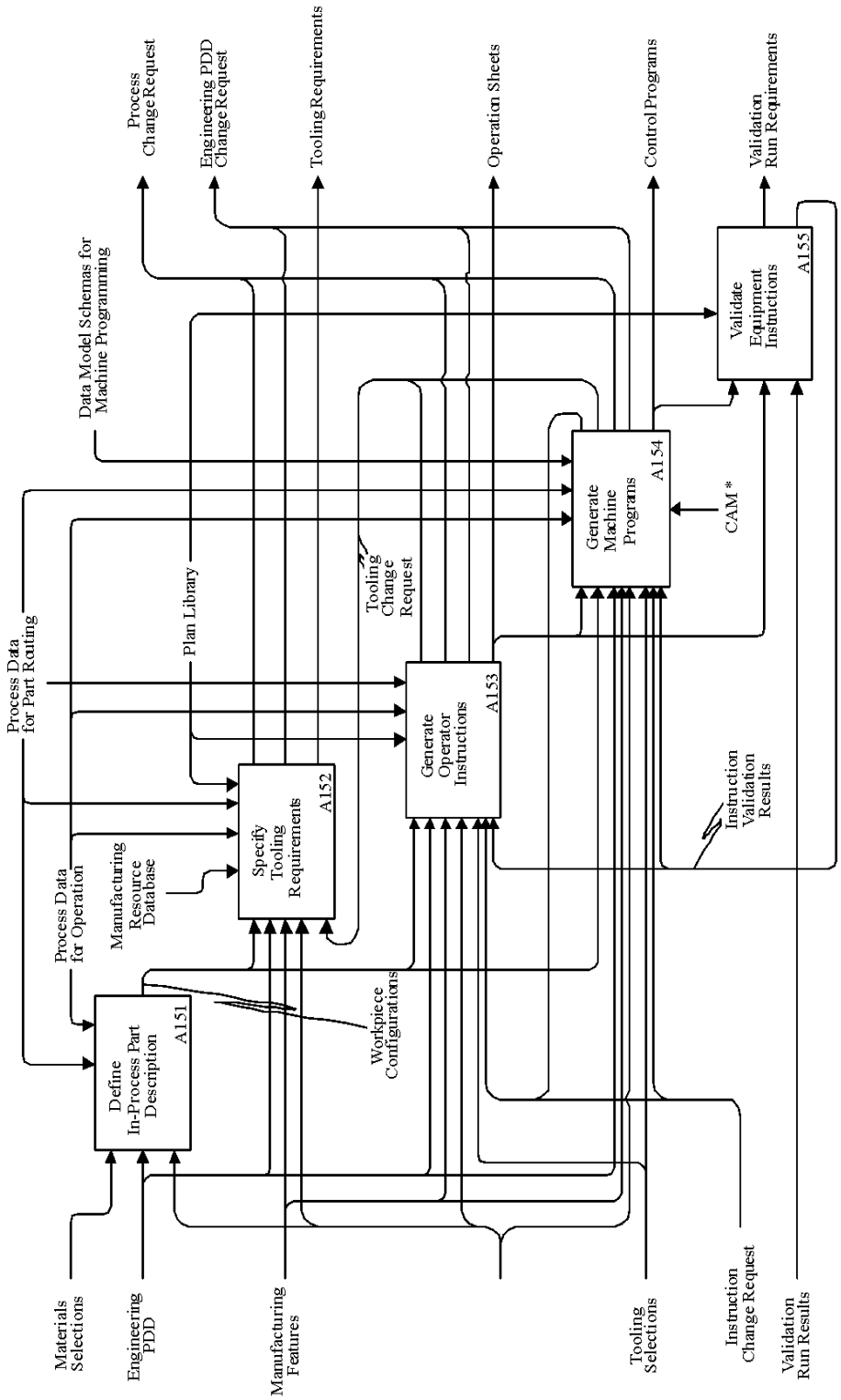


**Figure F.11 — A15 Develop Equipment Instructions**

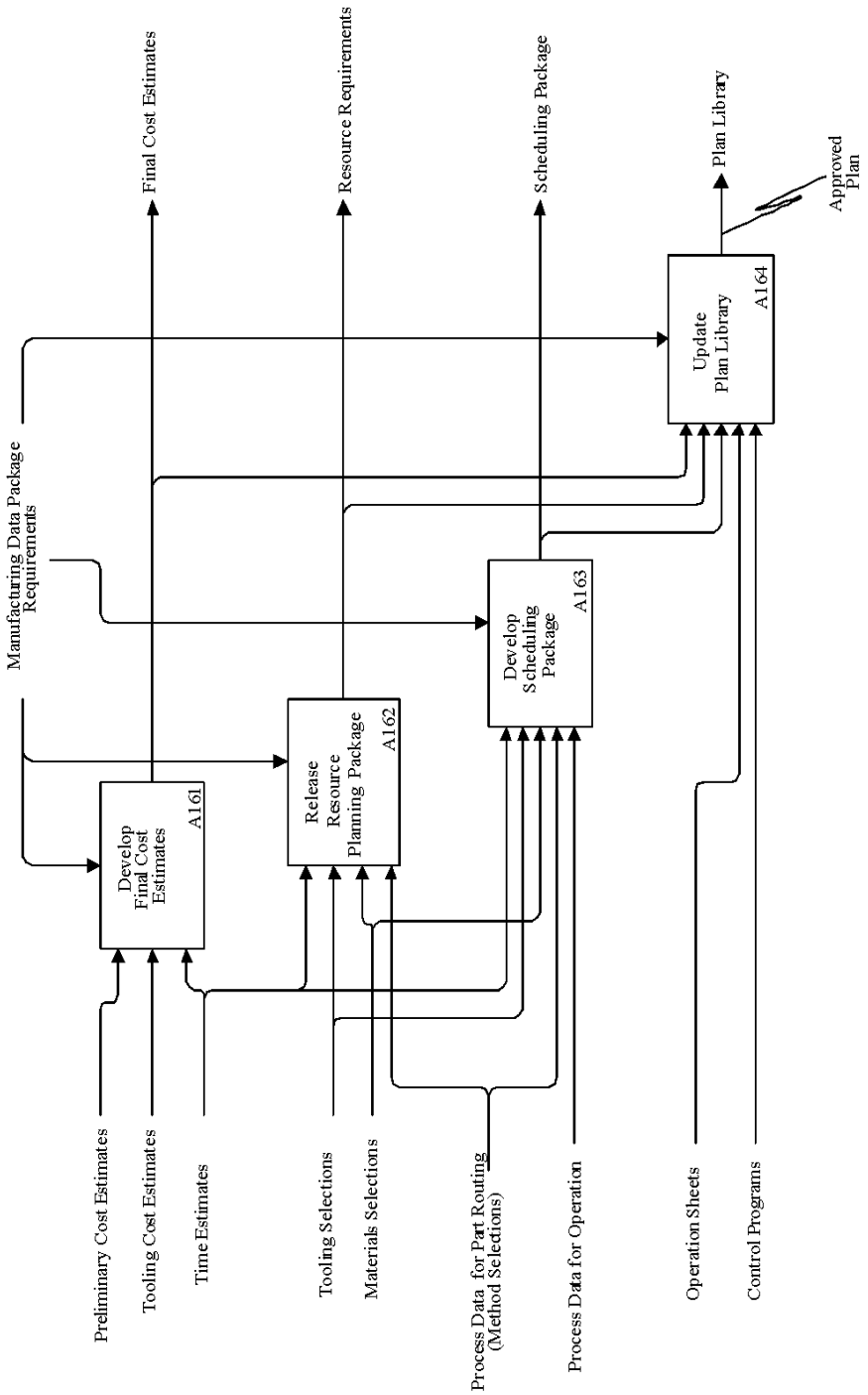


**Figure F.12 — A16 Finalize Manufacturing Data Package**

# Annex G
(informative)

# Application reference model

This Annex provides the application reference model for this part of ISO 10303 and is given in Figure G.1 to Figure G.20. 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 EXPRESS-G. The application reference model is independent of any implementation method. EXPRESS-G is defined in Annex D of ISO 10303-11.

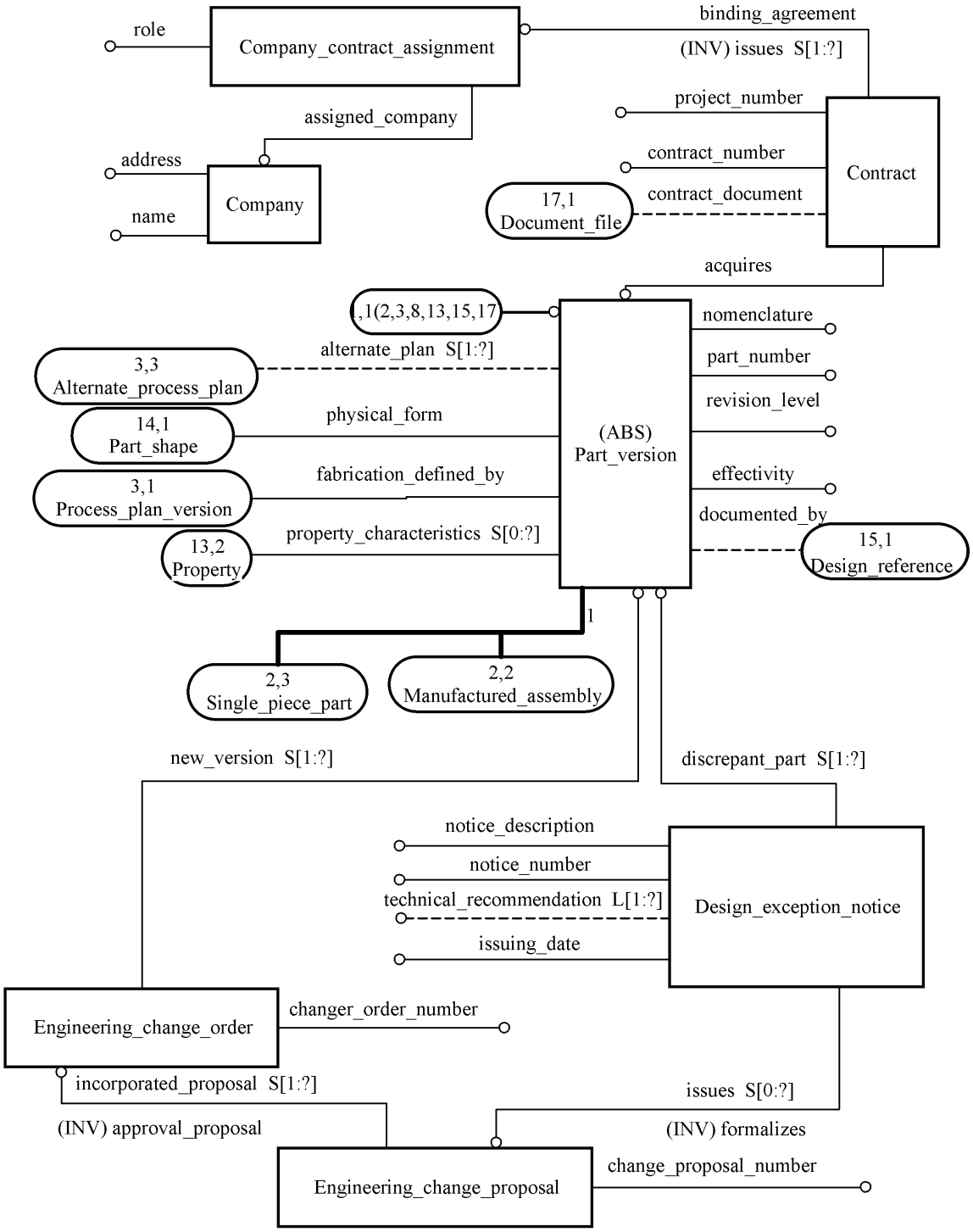


**Figure G.1 — ARM EXPRESS-G diagram 1 of 20**

 

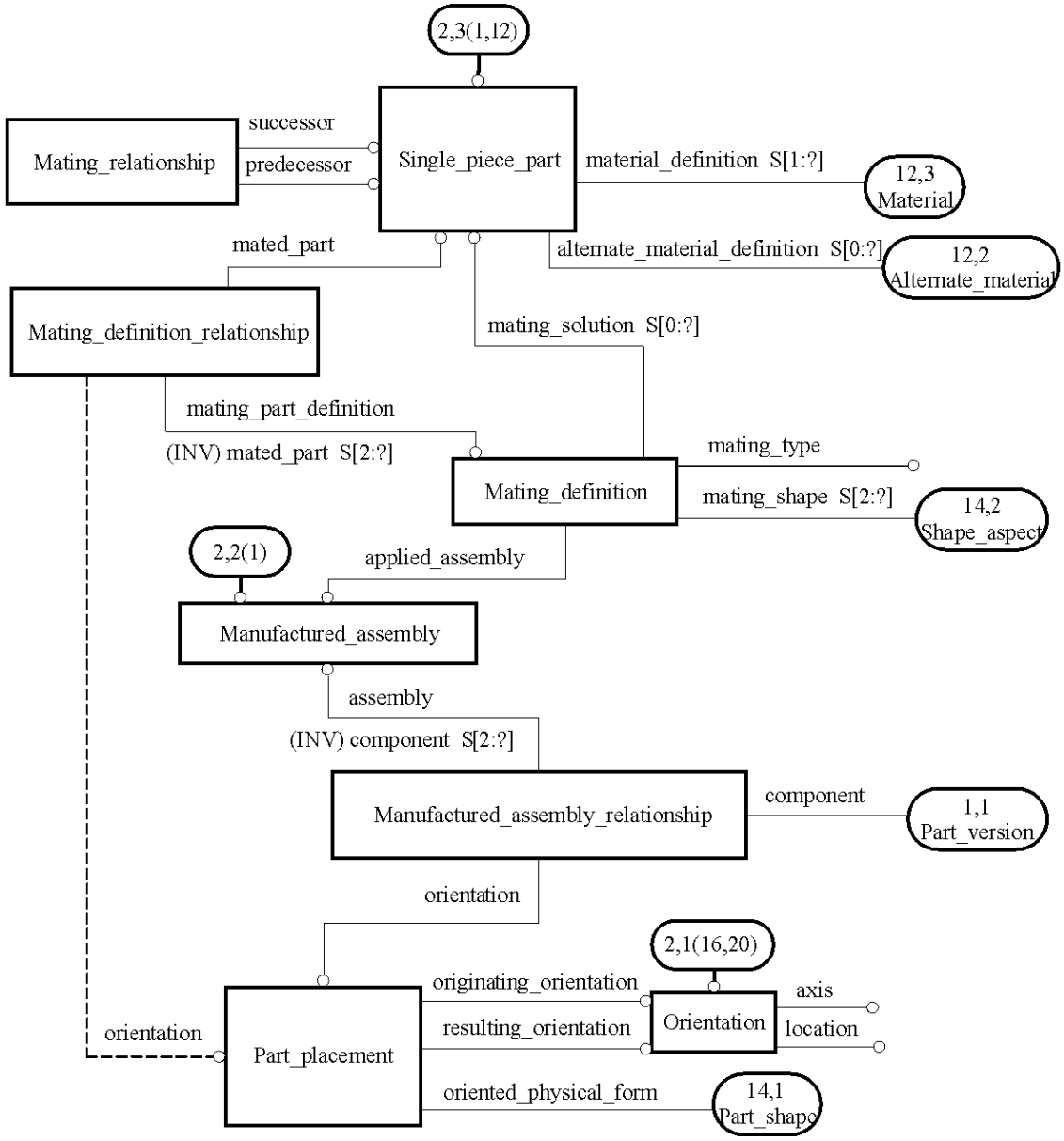


**Figure G.2 — ARM EXPRESS-G diagram 2 of 20**

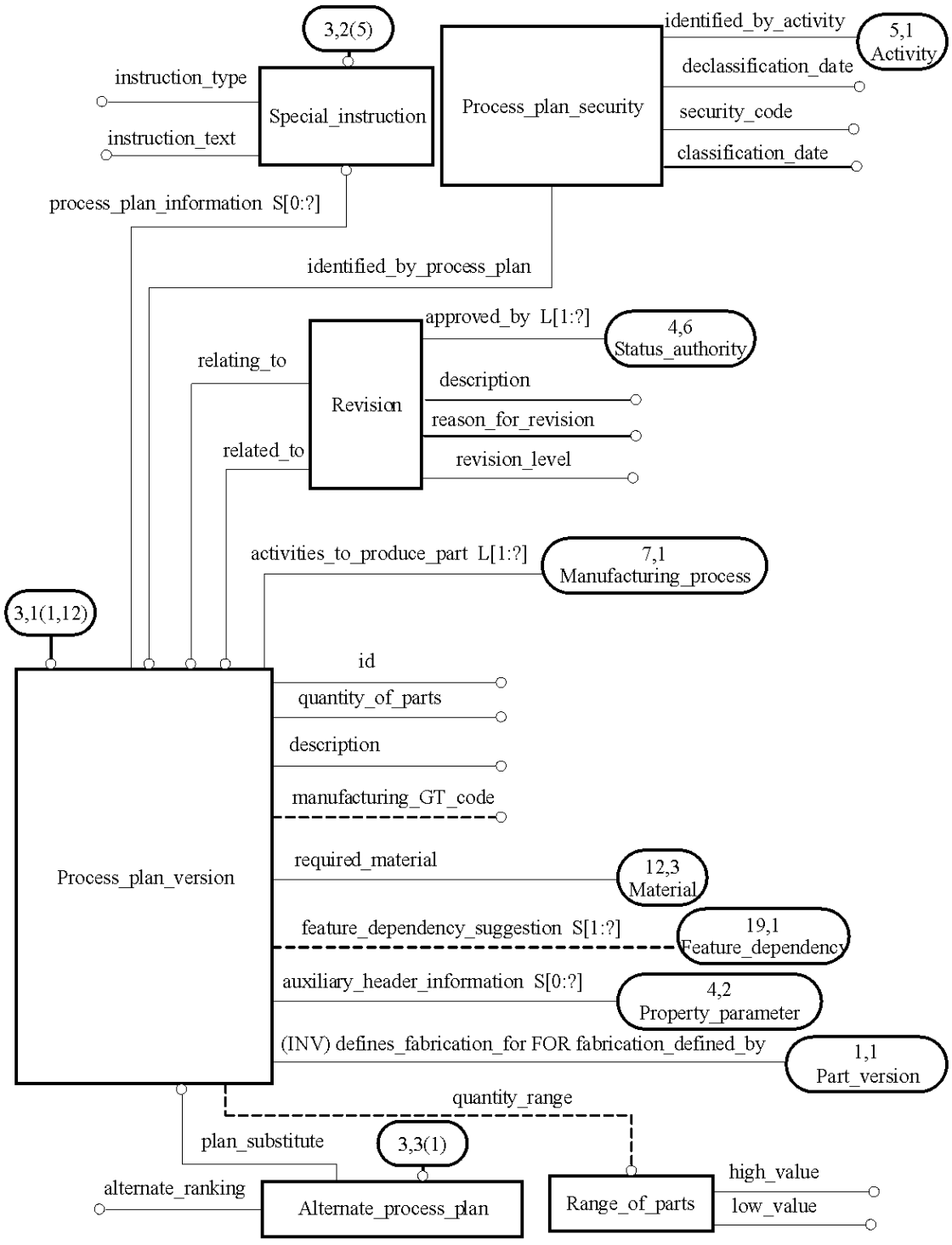


**Figure G.3 — ARM EXPRESS-G diagram 3 of 20**

 

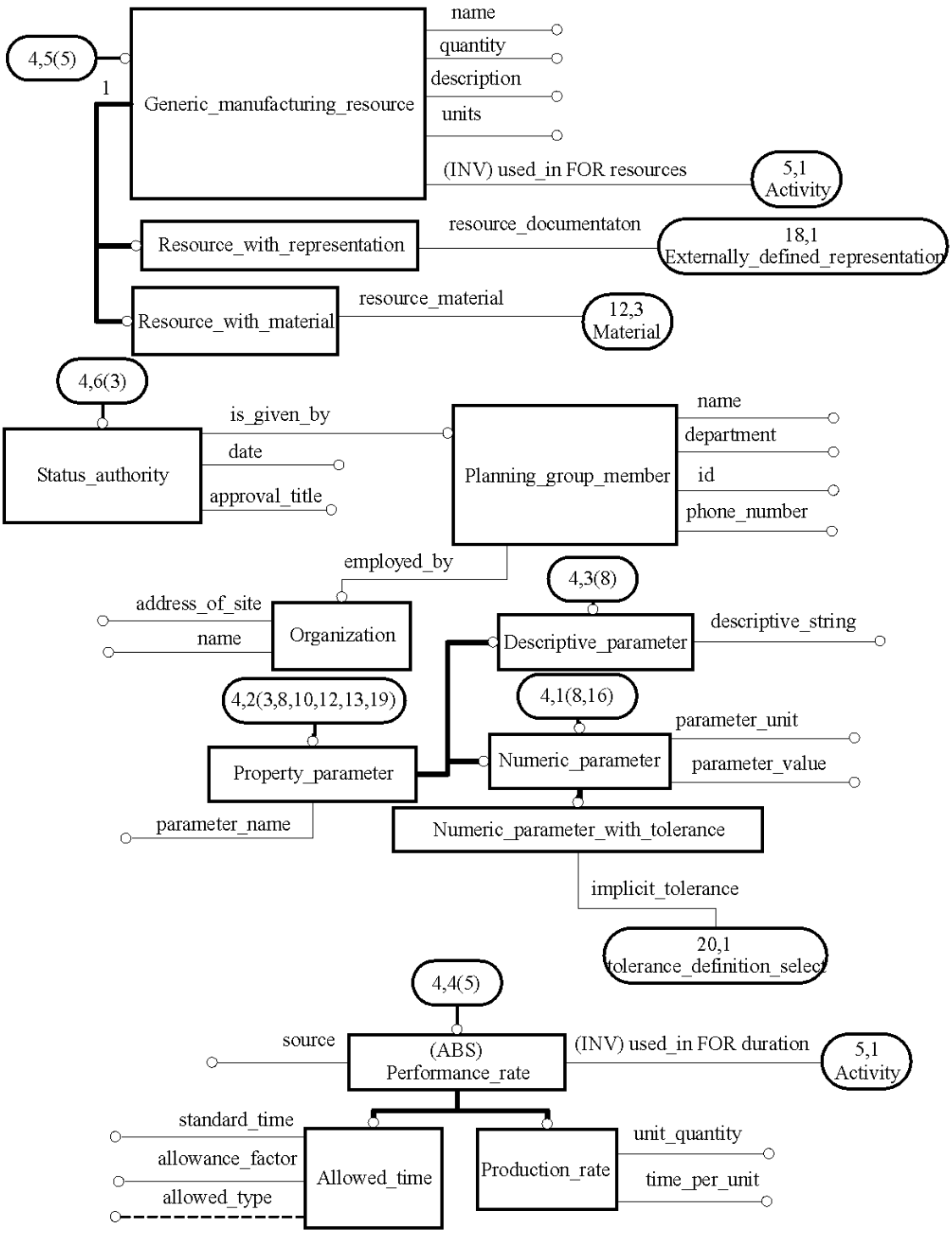


**Figure G.4 — ARM EXPRESS-G diagram 4 of 20**

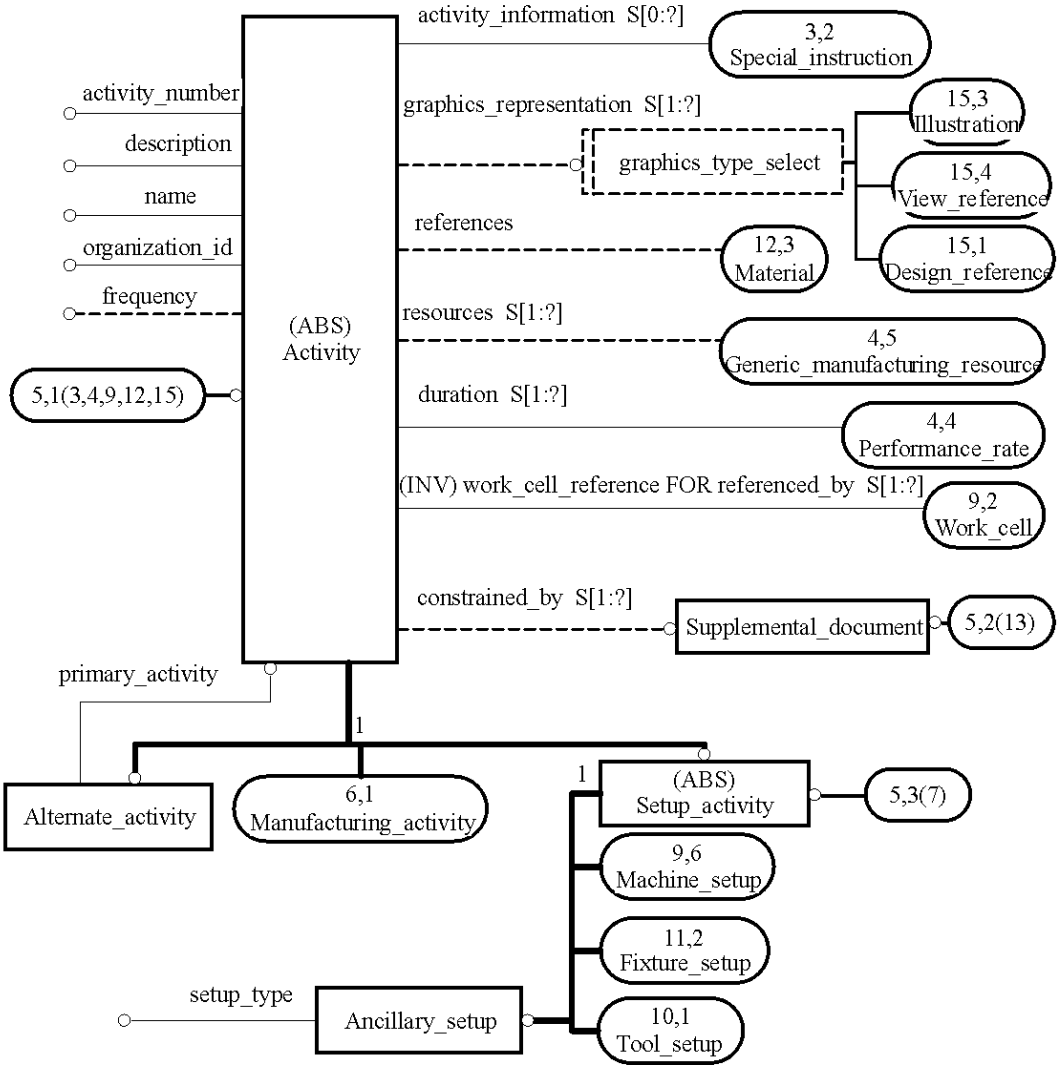


**Figure G.5 — ARM EXPRESS-G diagram 5 of 20**

 

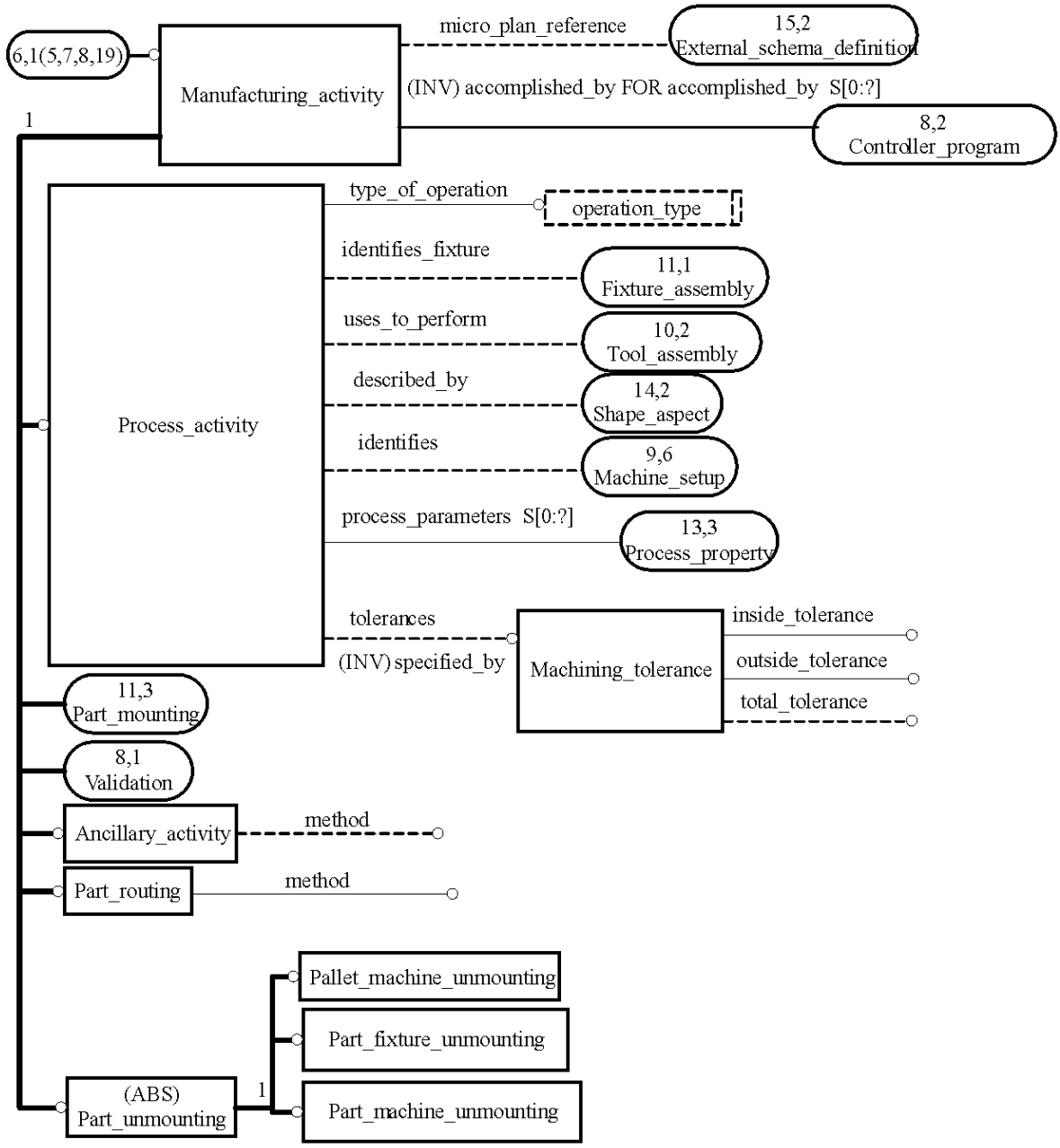


**Figure G.6 — ARM EXPRESS-G diagram 6 of 20**

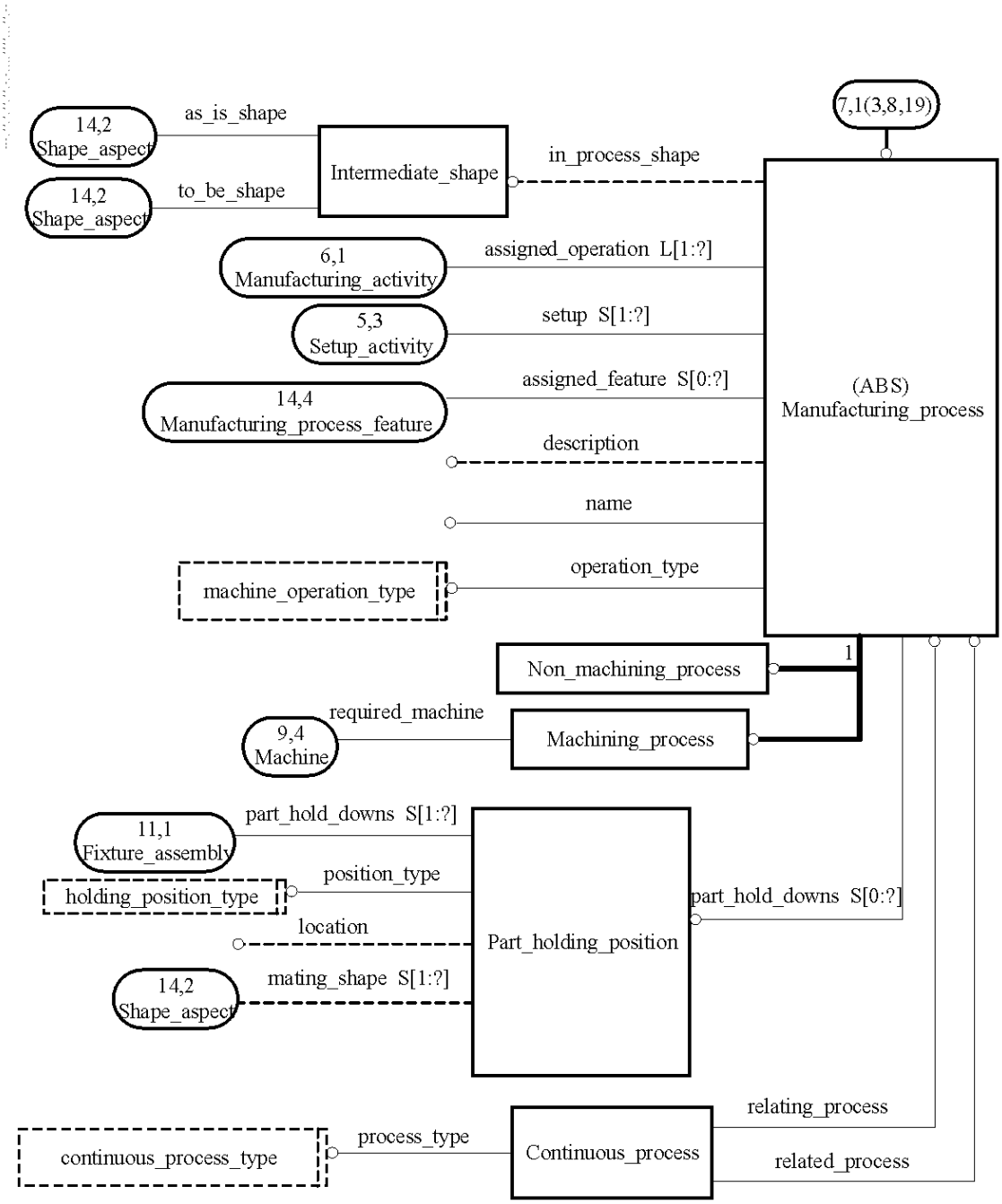


**Figure G.7 — ARM EXPRESS-G diagram 7 of 20**

 

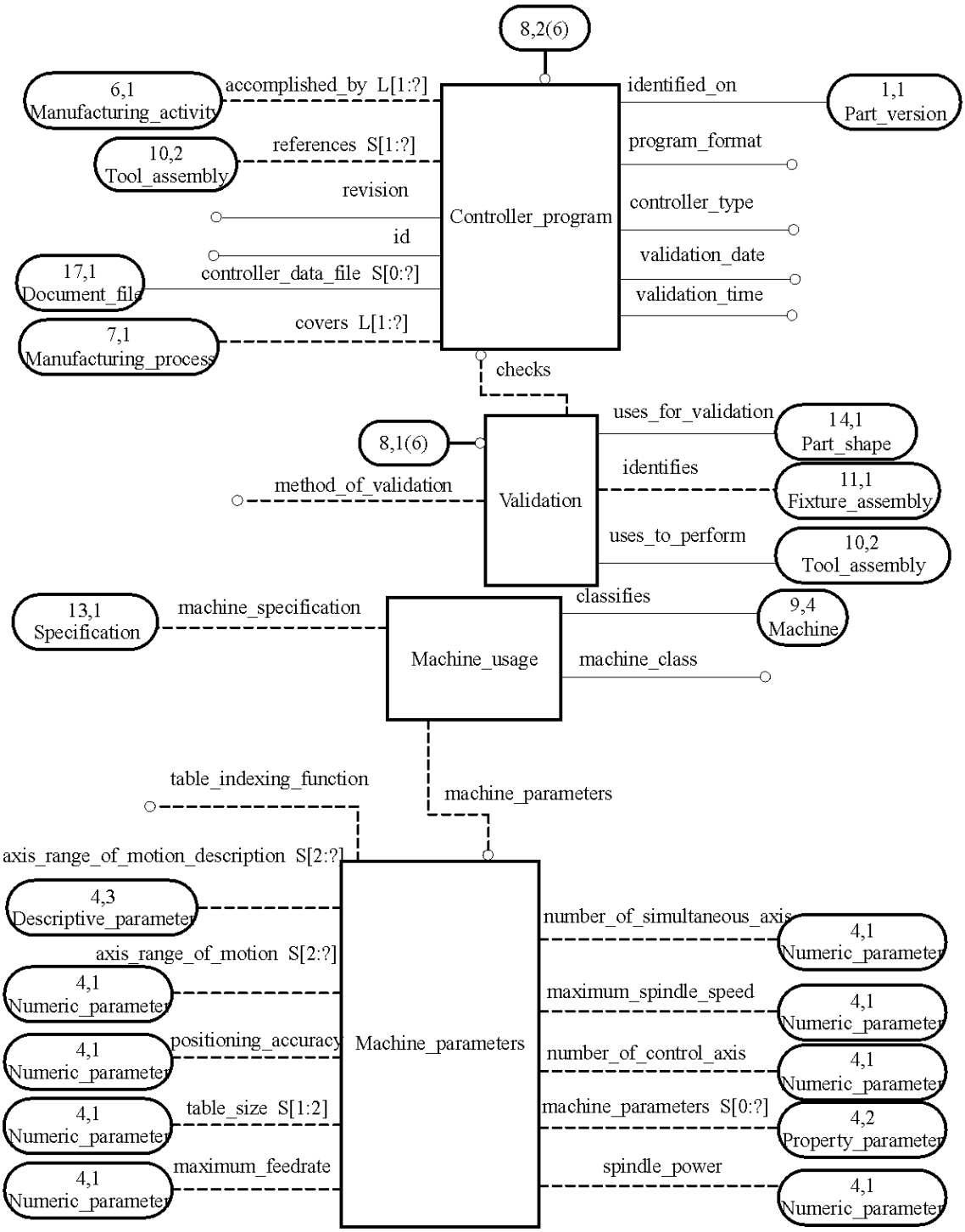


**Figure G.8 — ARM EXPRESS-G diagram 8 of 20**

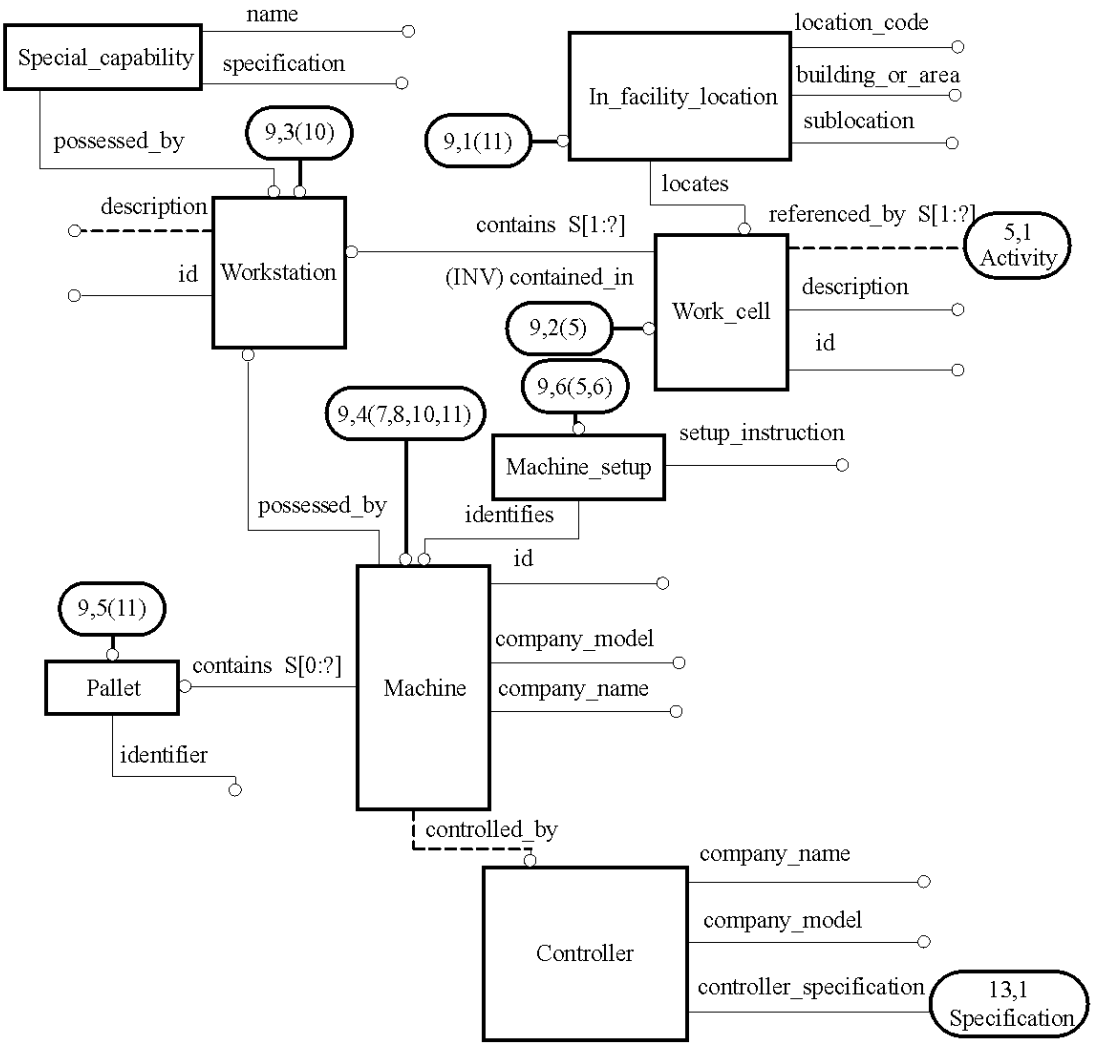


**Figure G.9 — ARM EXPRESS-G diagram 9 of 20**

 

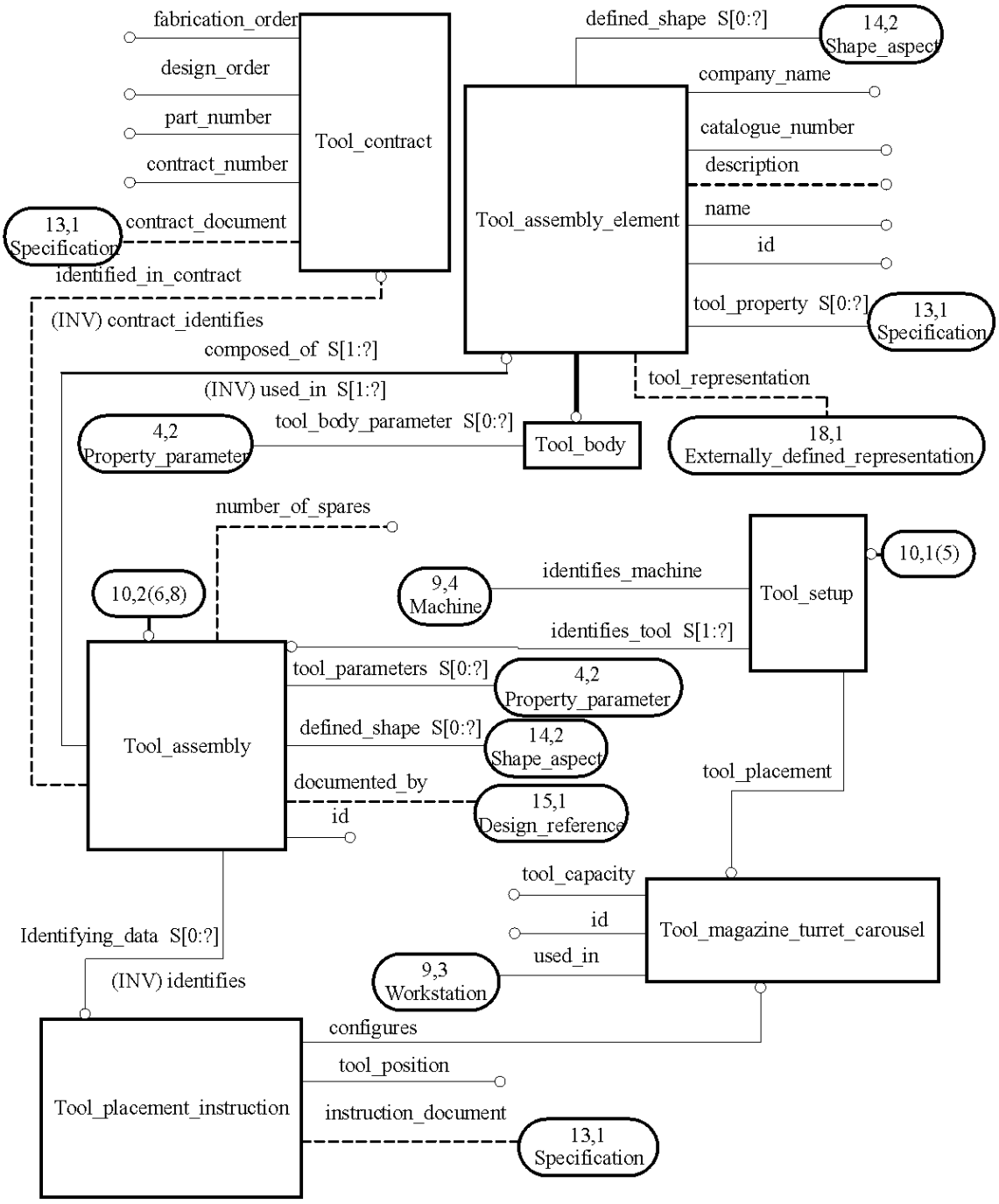


**Figure G.10 — ARM EXPRESS-G diagram 10 of 20**

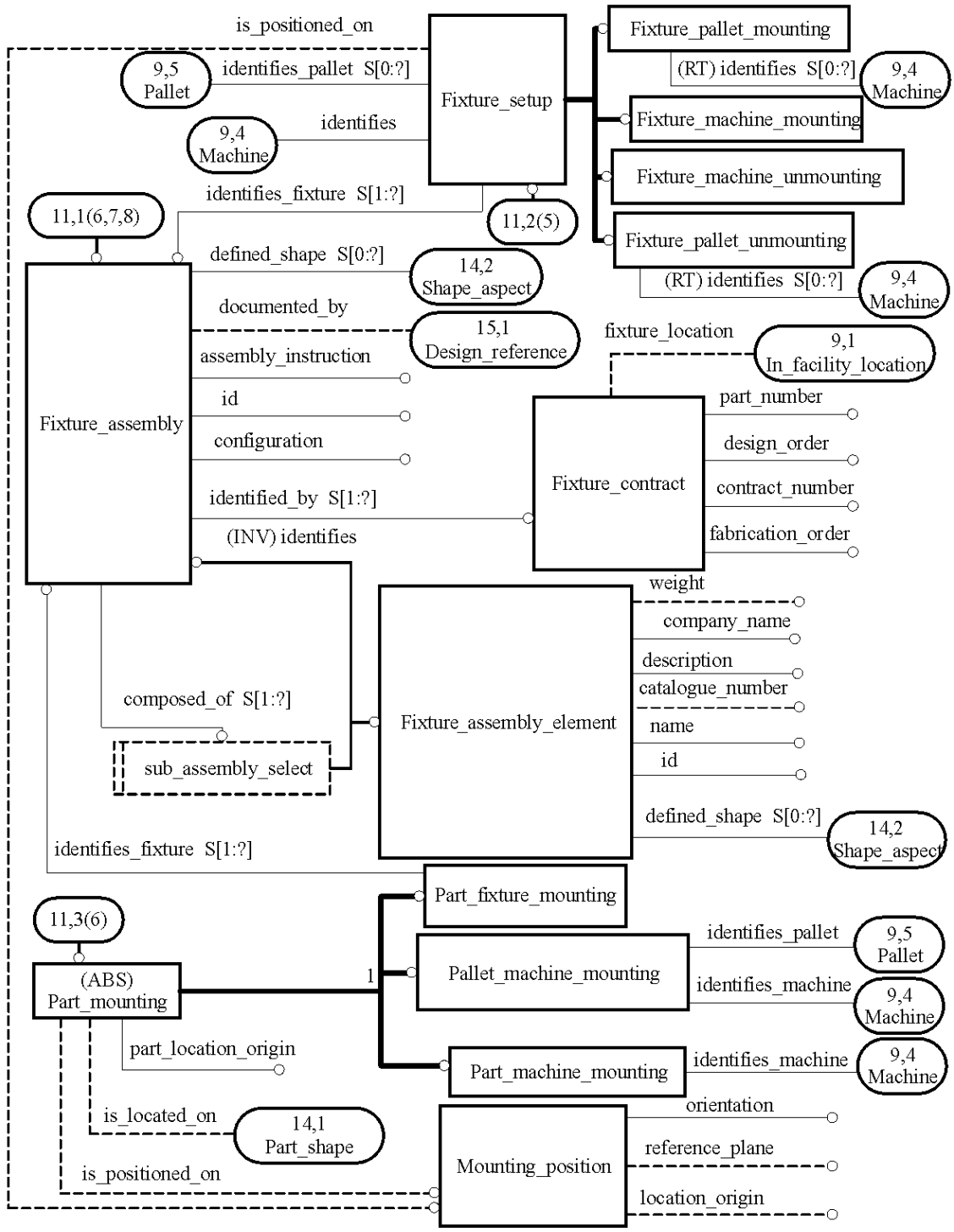


**Figure G.11 — ARM EXPRESS-G diagram 11 of 20**

 

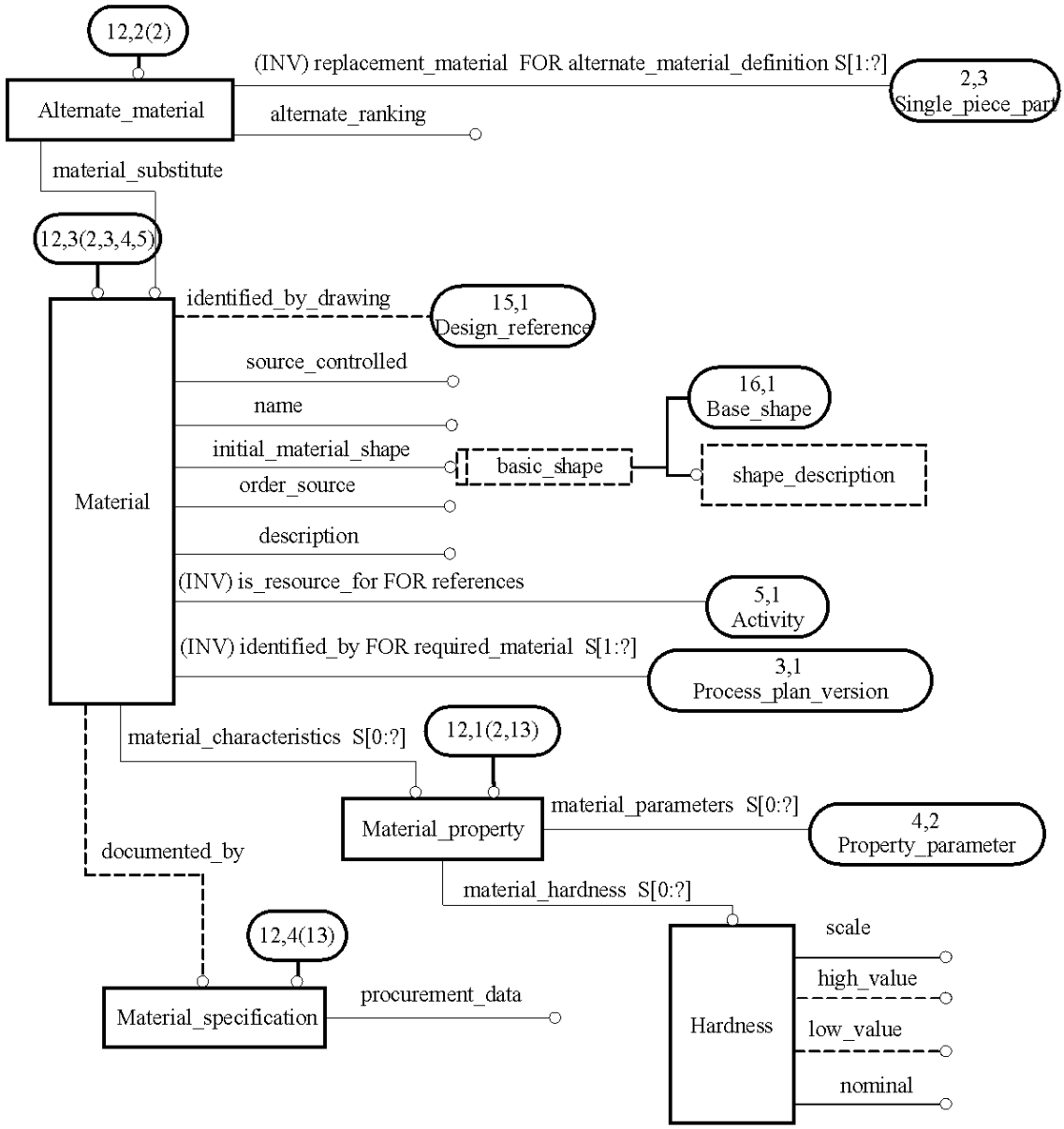


**Figure G.12 — ARM EXPRESS-G diagram 12 of 20**

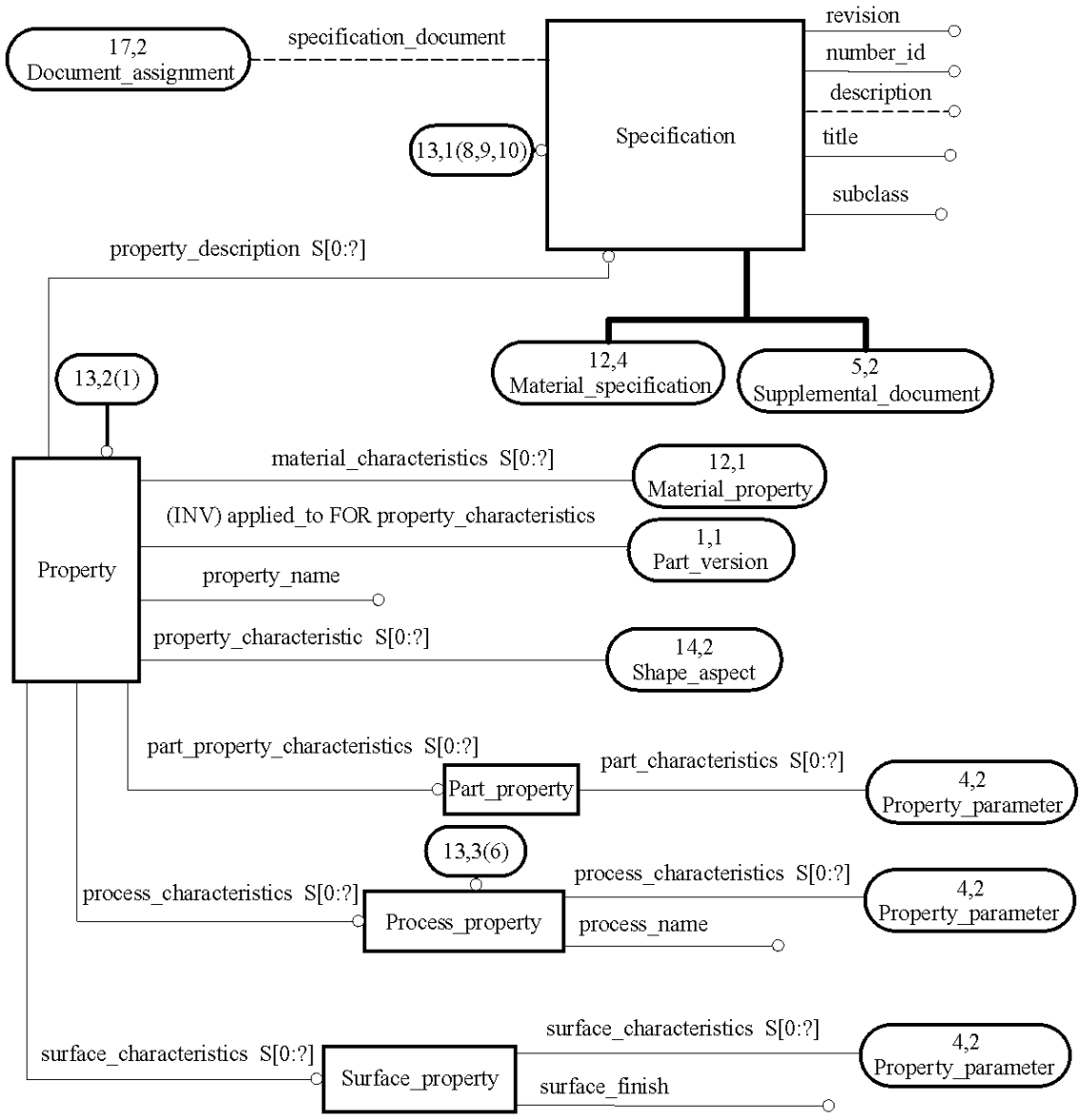


**Figure G.13 — ARM EXPRESS-G diagram 13 of 20**

 

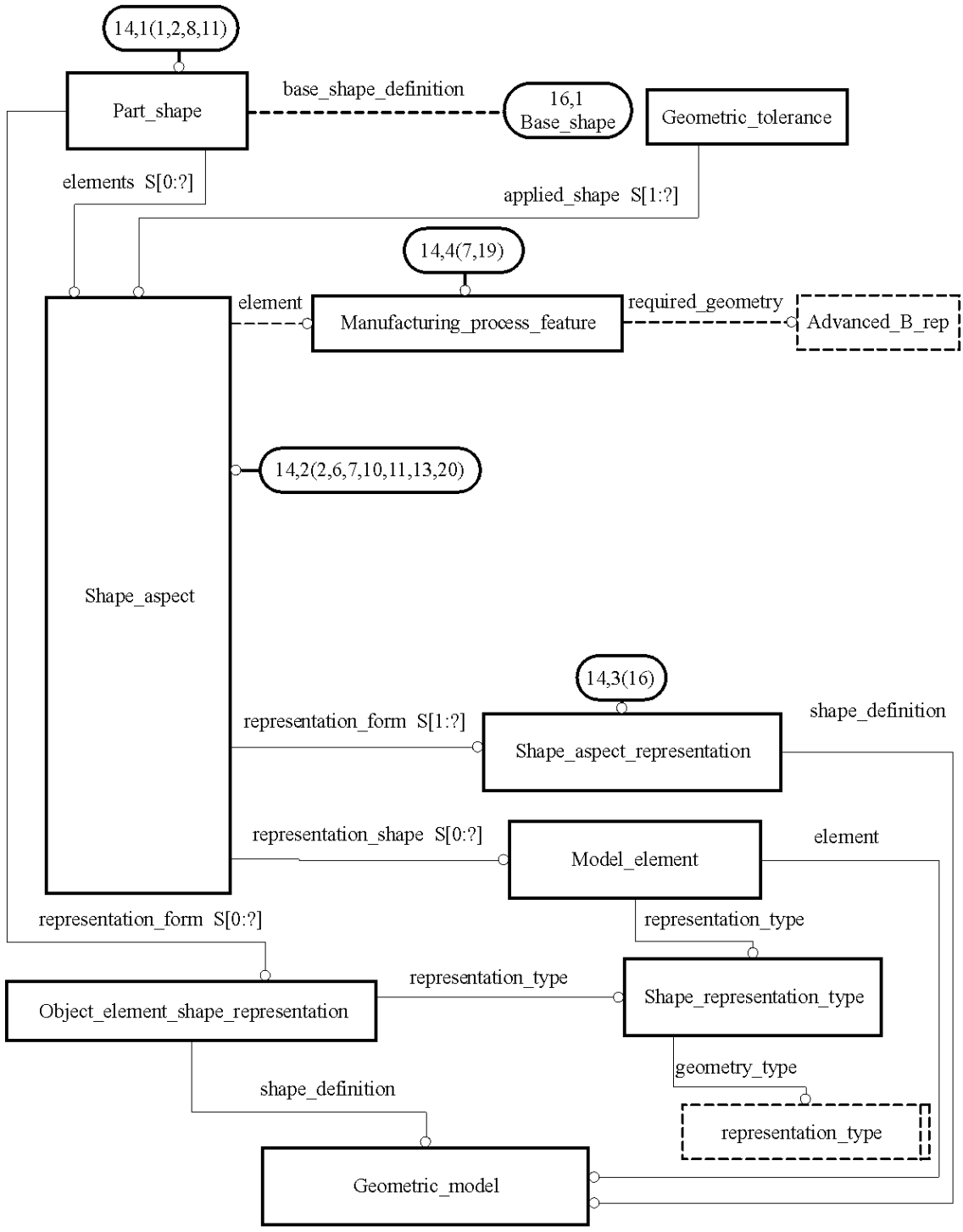


**Figure G.14 — ARM EXPRESS-G diagram 14 of 20**

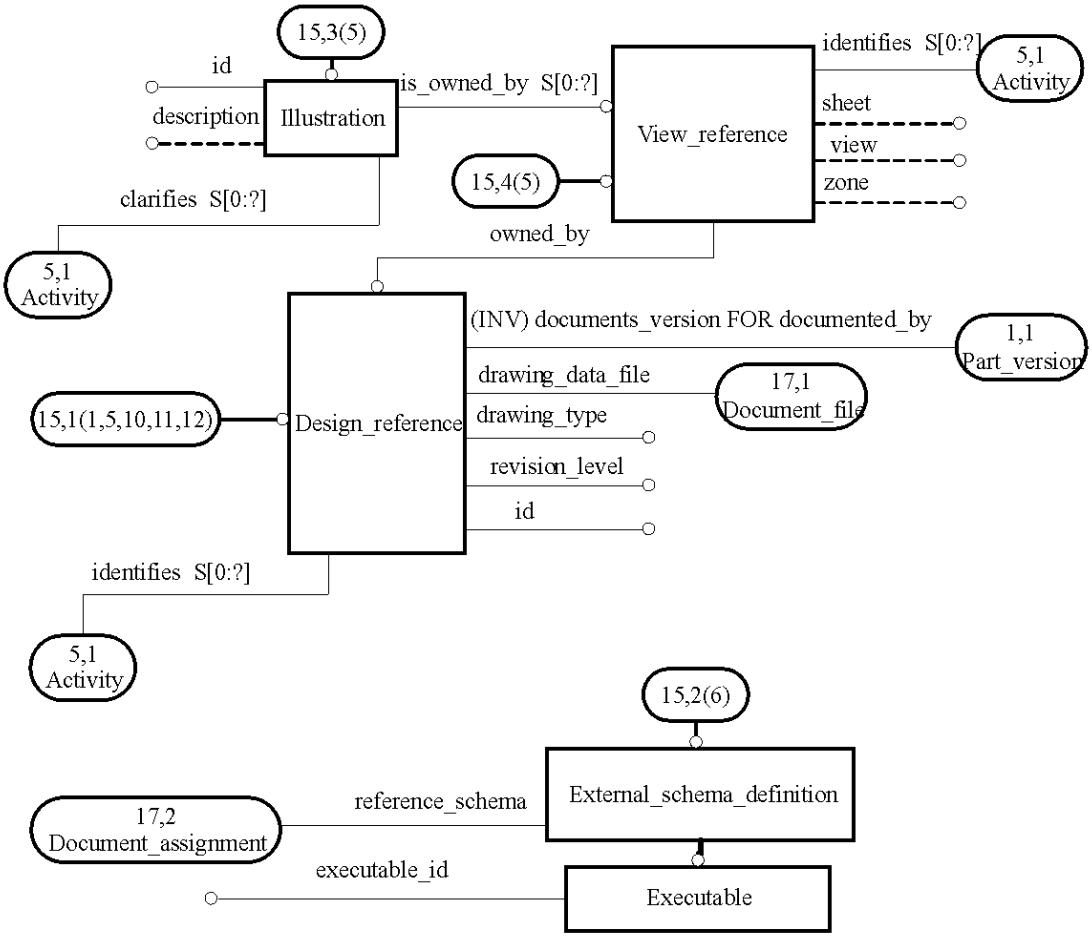


**Figure G.15 — ARM EXPRESS-G diagram 15 of 20**

 

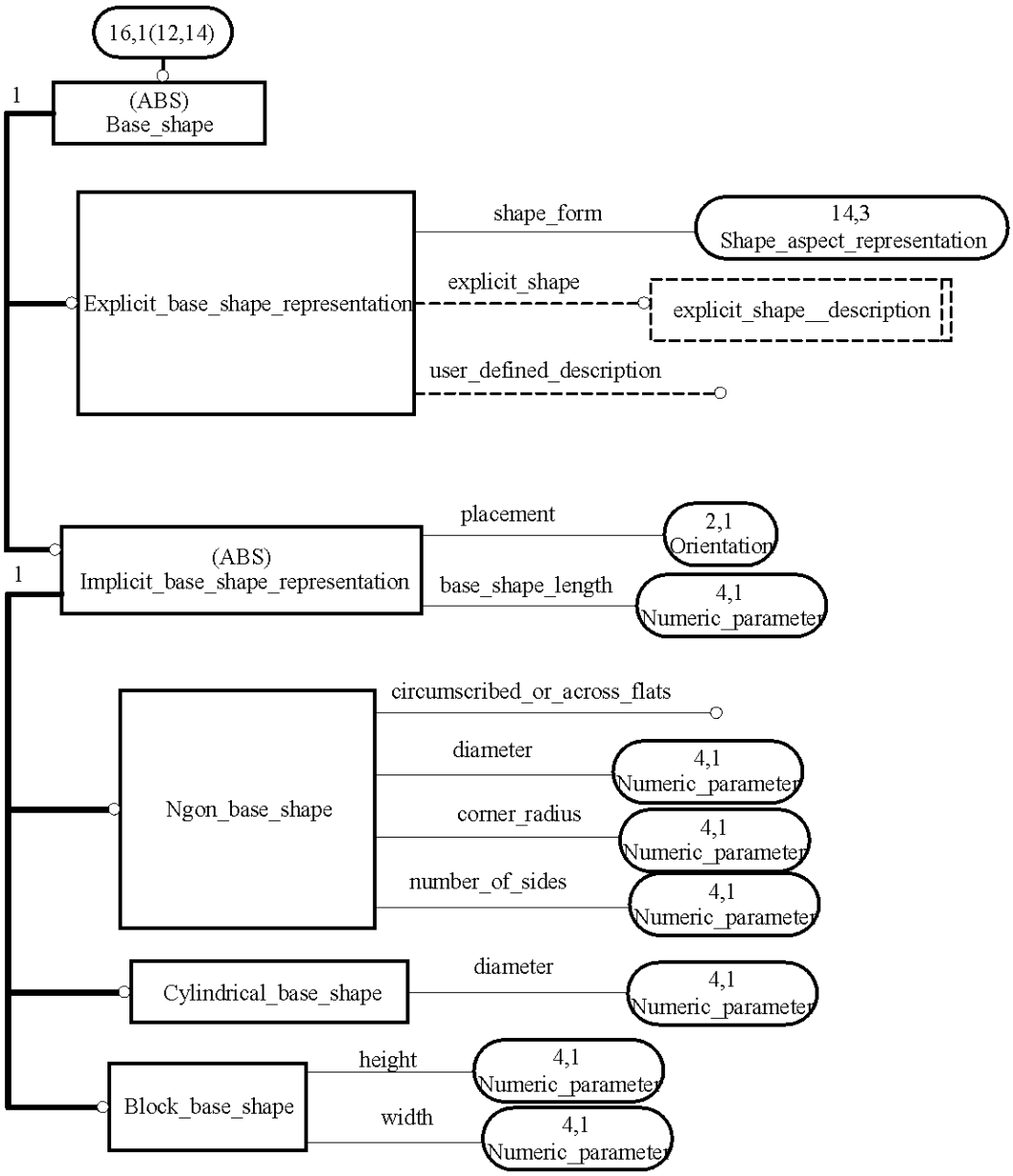


**Figure G.16 — ARM EXPRESS-G diagram 16 of 20**

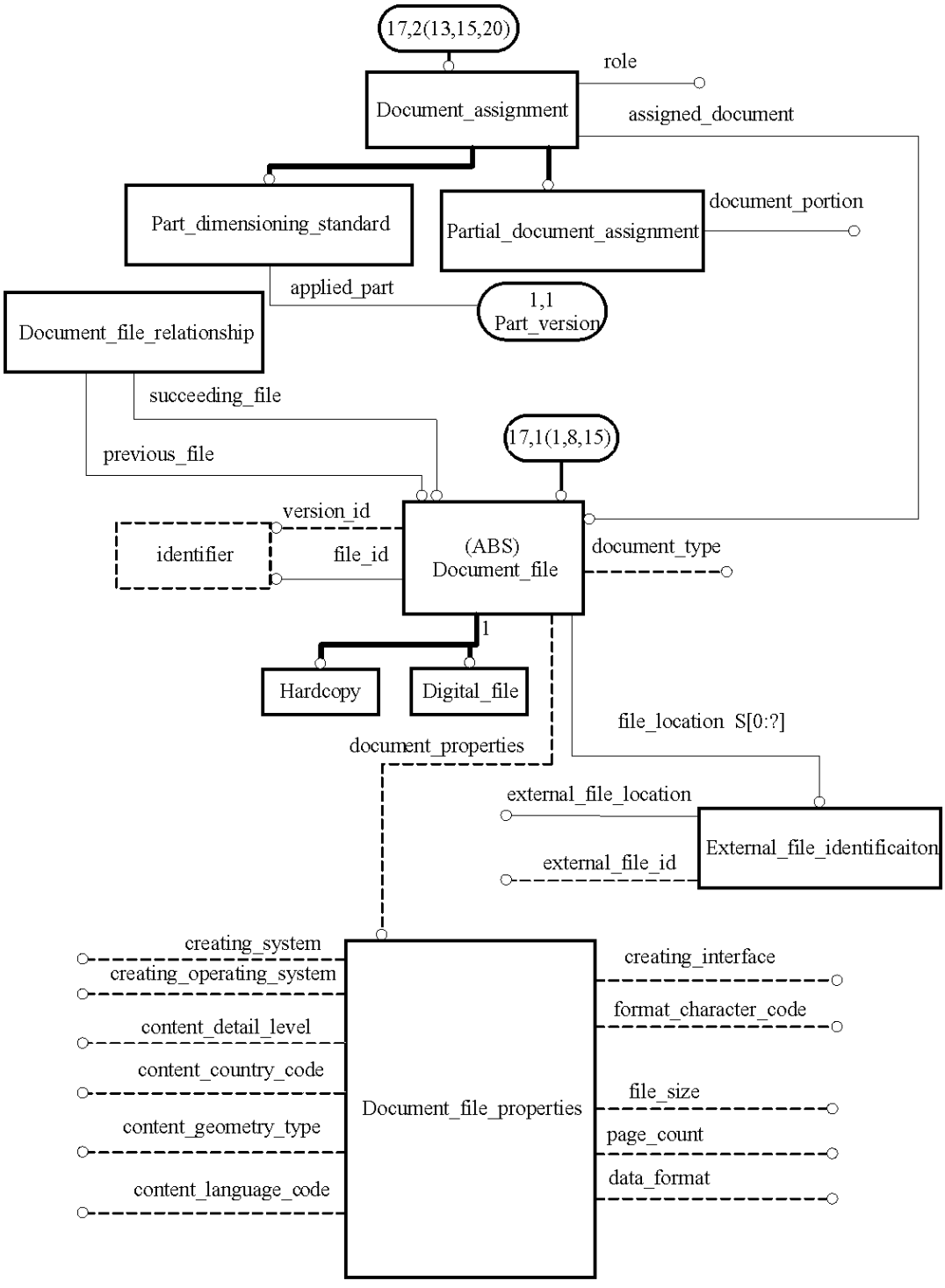


**Figure G.17 — ARM EXPRESS-G diagram 17 of 20**

 

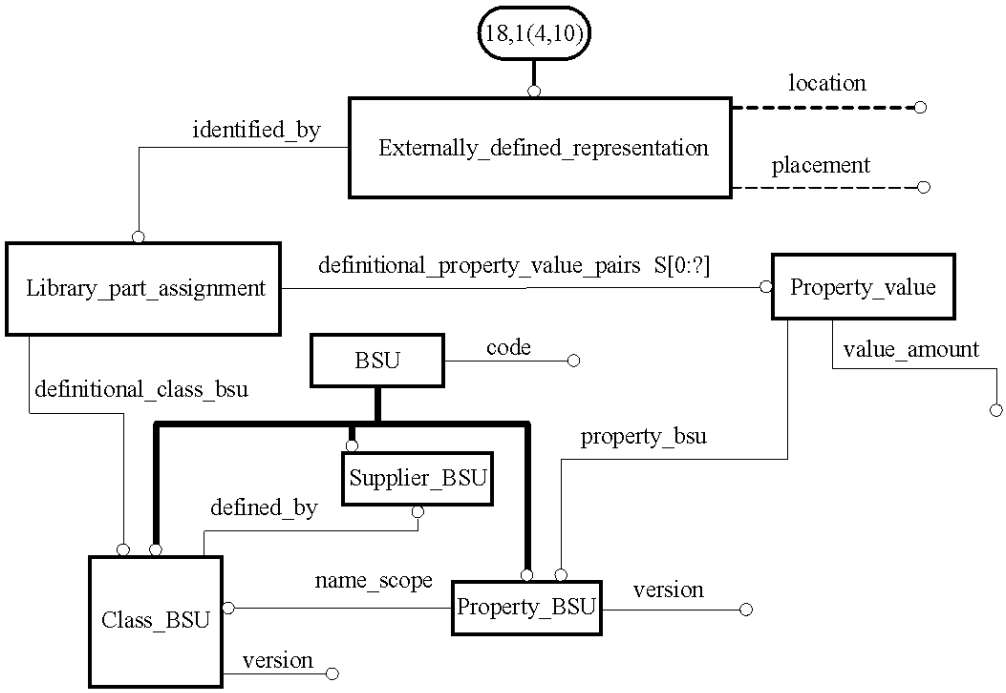


**Figure G.18 — ARM EXPRESS-G diagram 18 of 20**

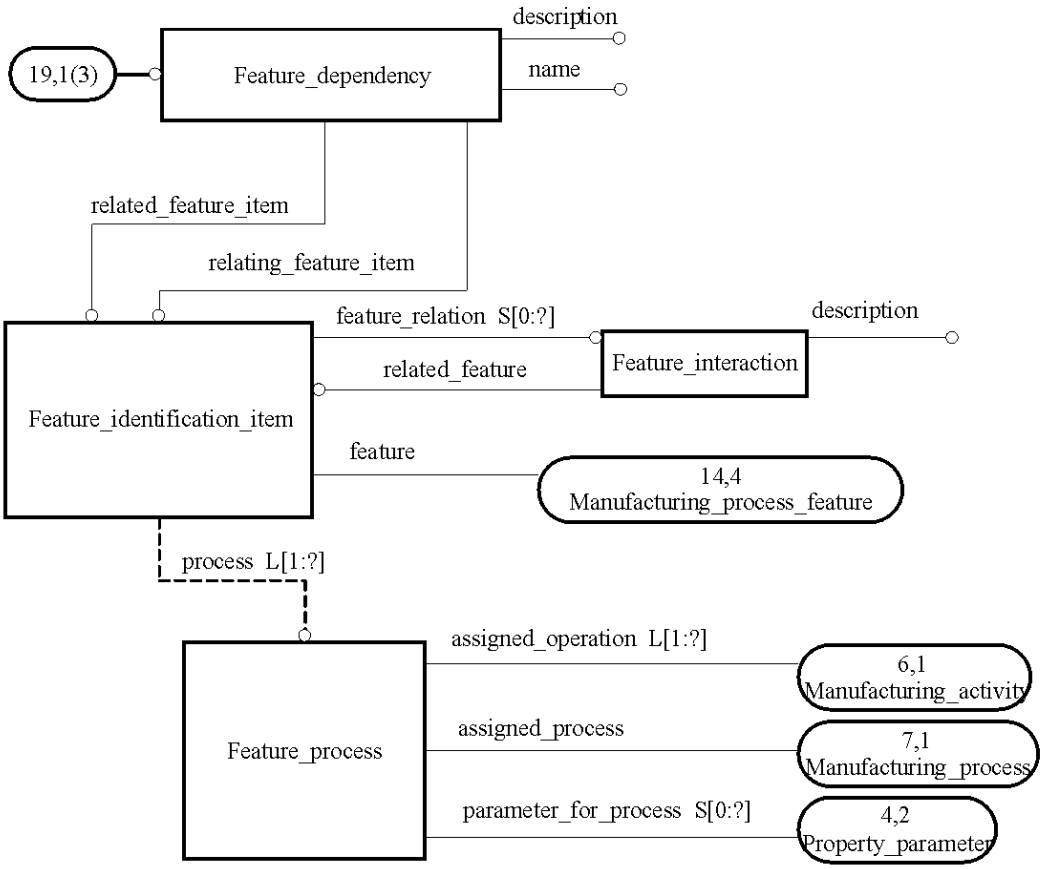


**Figure G.19 — ARM EXPRESS-G diagram 19 of 20**

 

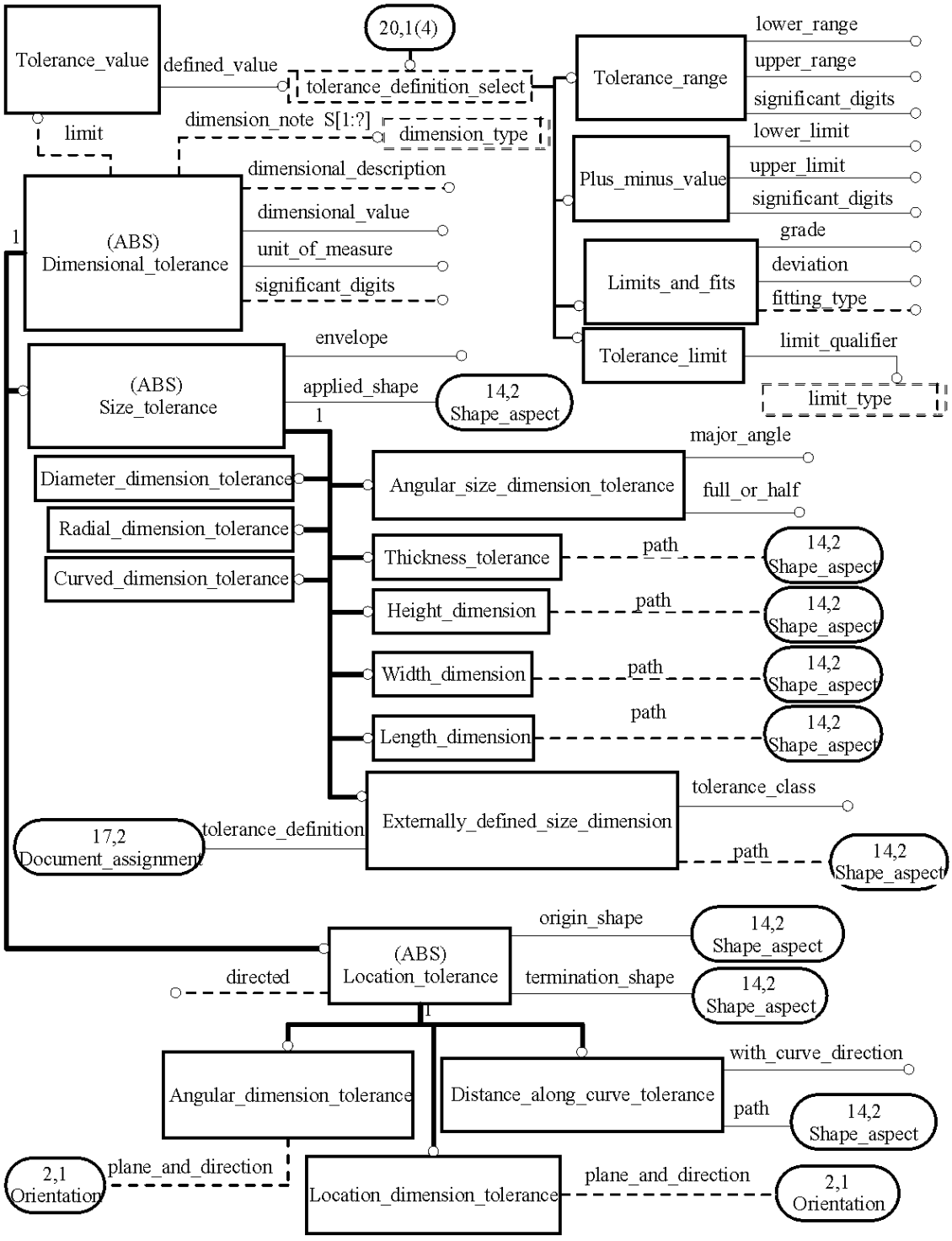


**Figure G.20 — ARM EXPRESS-G diagram 20 of 20**

# Annex H
(informative)

# AIM EXPRESS-G

Figure H.1 through H.39 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 an Annex A of ISO 10303-11.

 

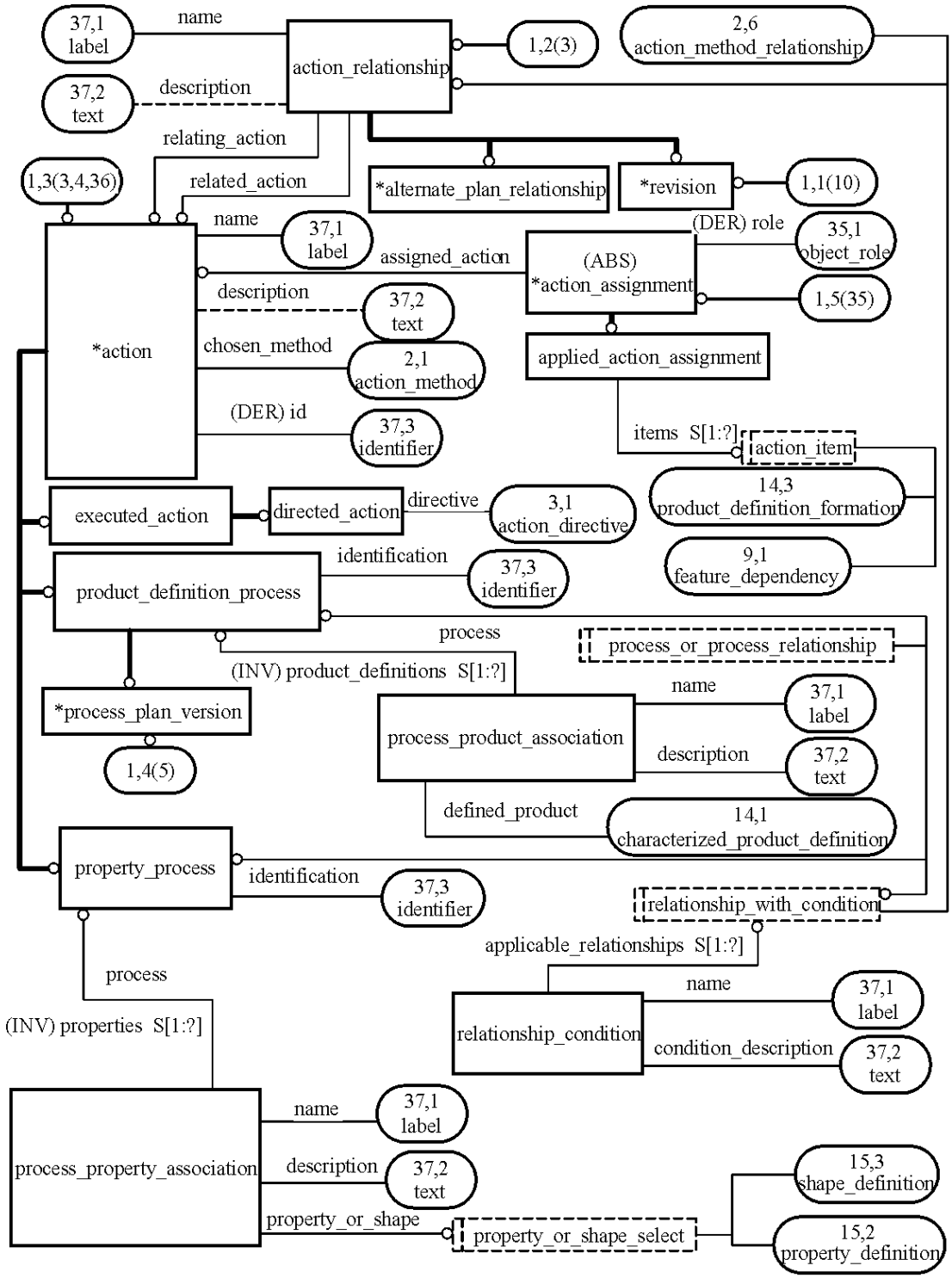


**Figure H.1 — action - AIM EXPRESS-G diagram 1 of 36**

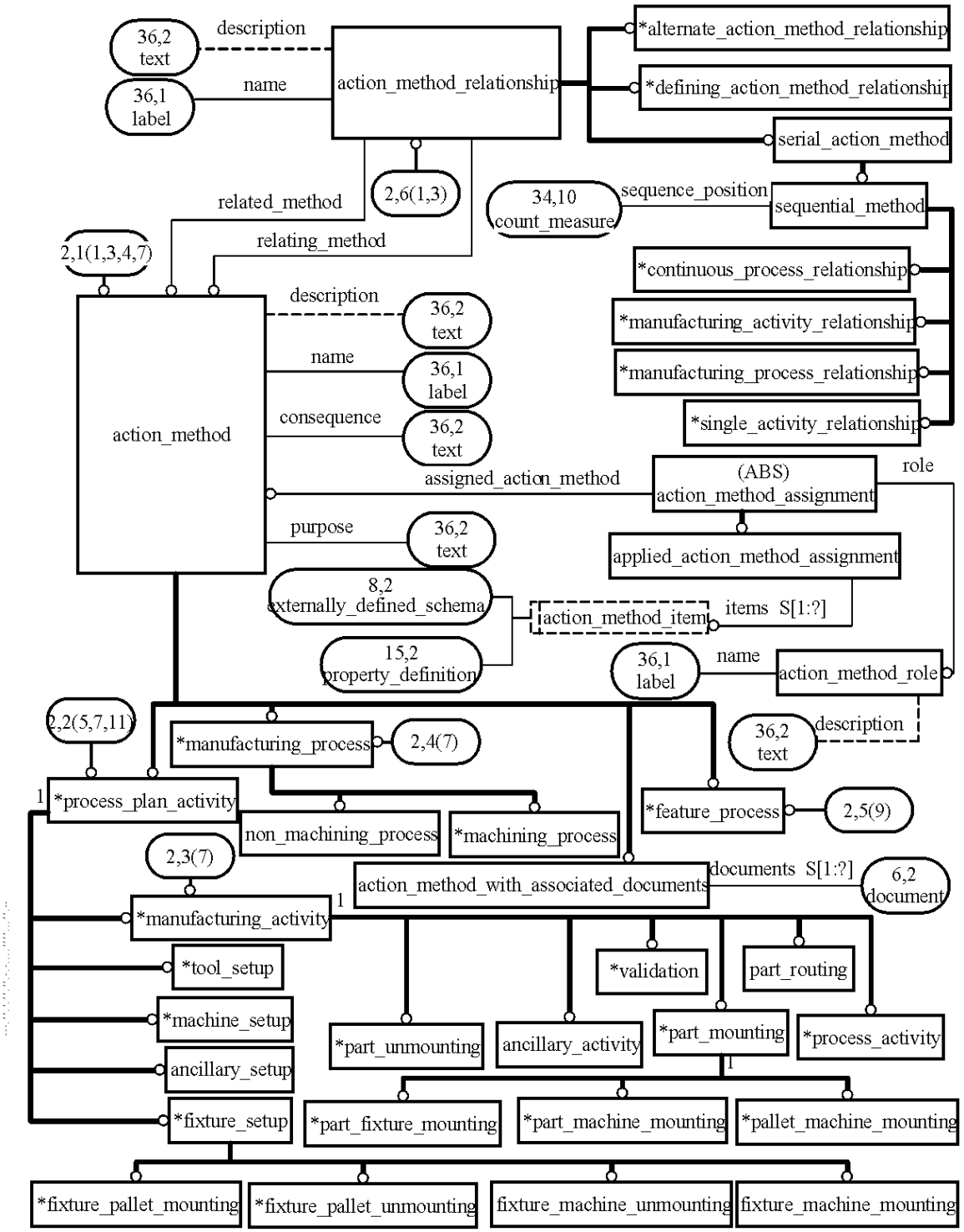


**Figure H.2 — action_method - AIM EXPRESS-G diagram 2 of 36**

 

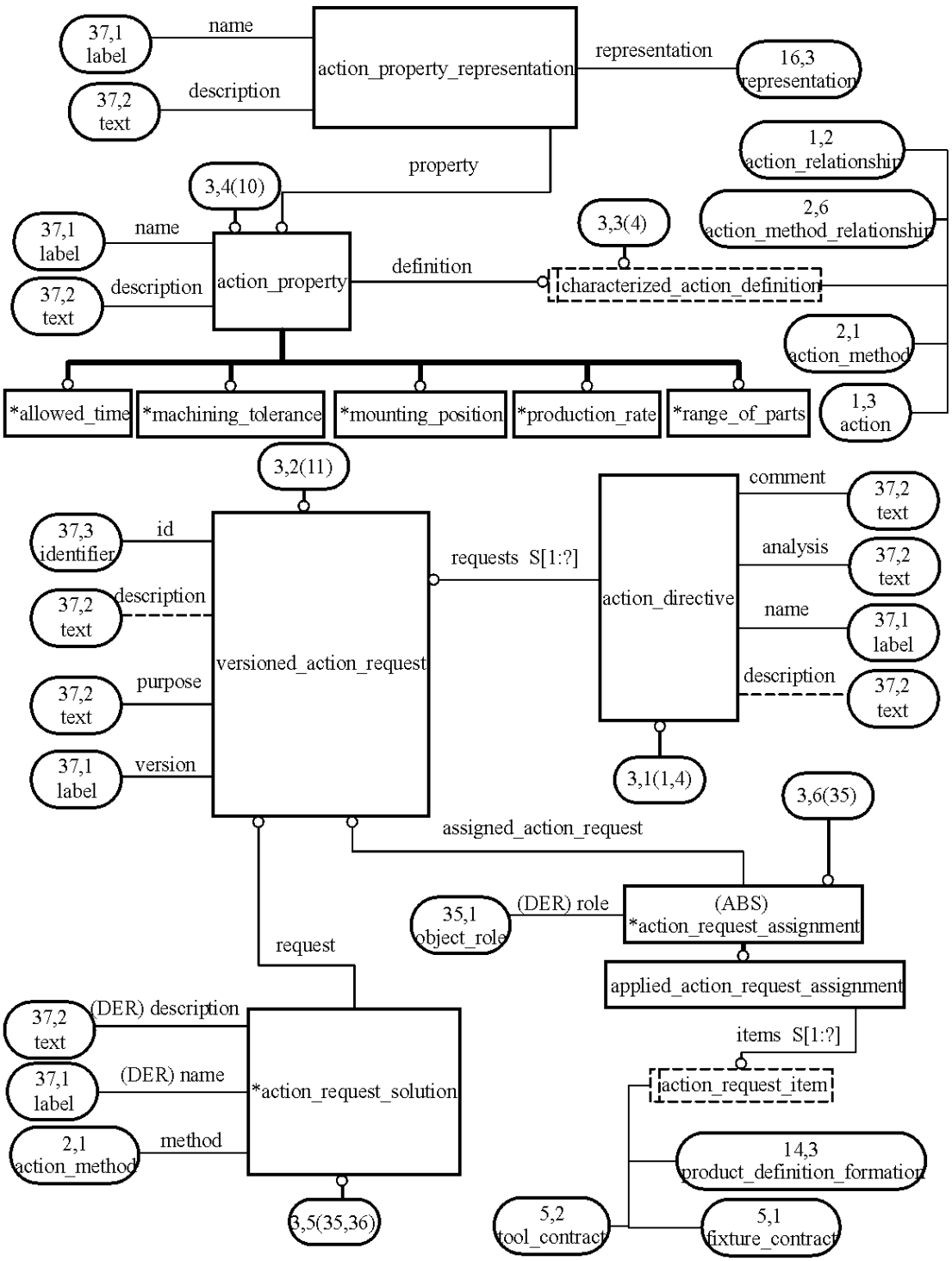


**Figure H.3 — action_property - AIM EXPRESS-G diagram 3 of 36**

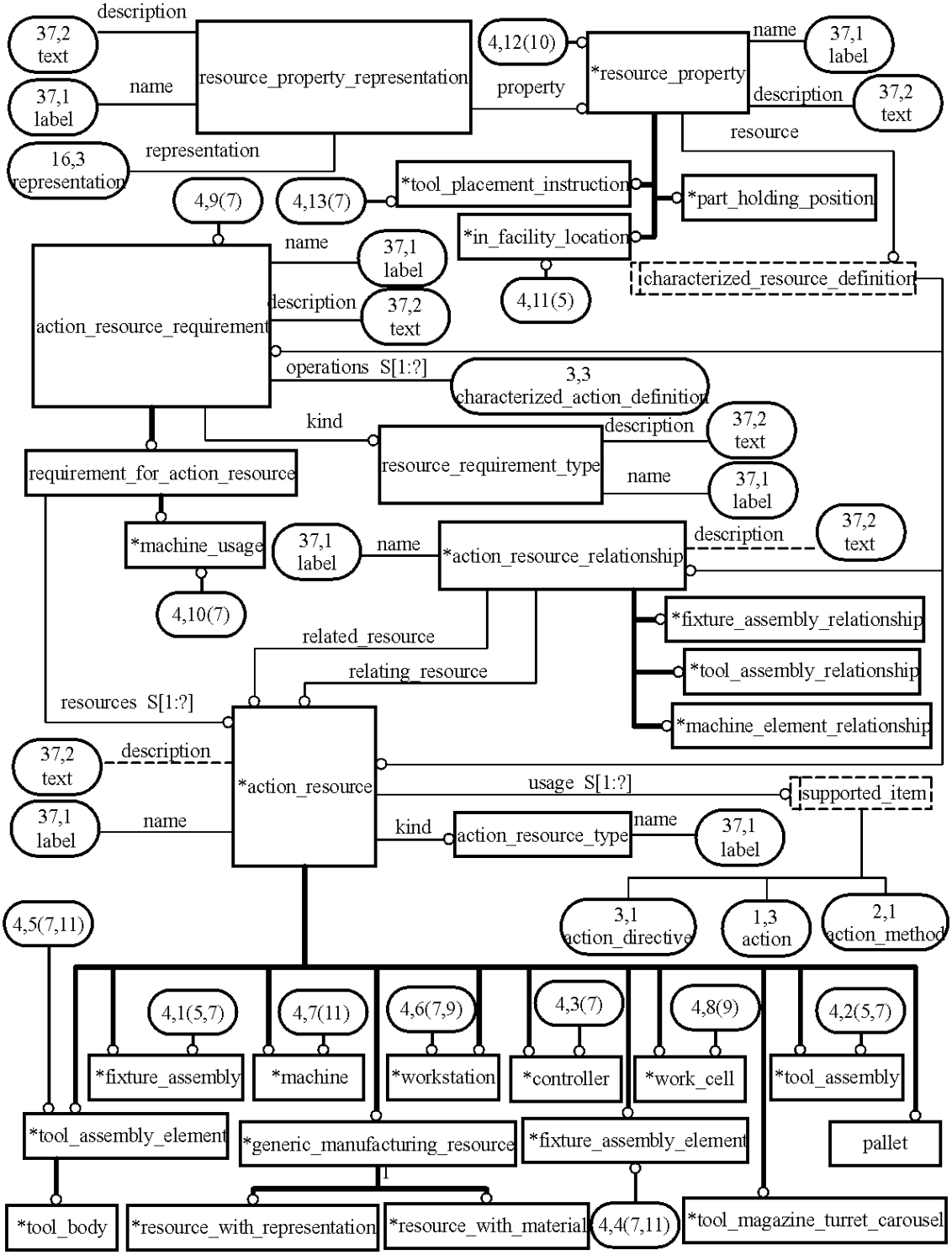


**Figure H.4 — action_resource - AIM EXPRESS-G diagram 4 of 36**

 

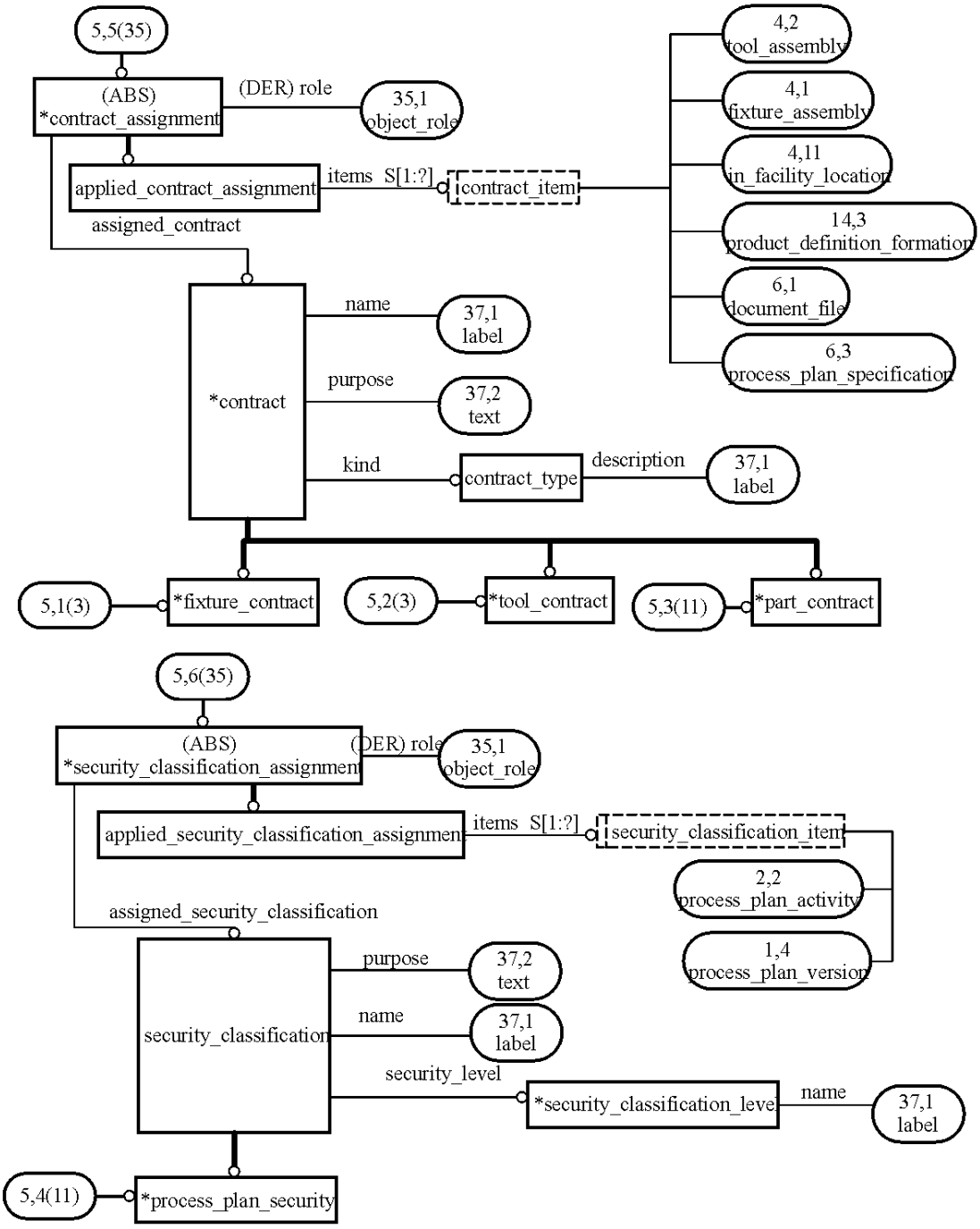


**Figure H.5 — contract and security_classification - AIM EXPRESS-G diagram 5 of 36**

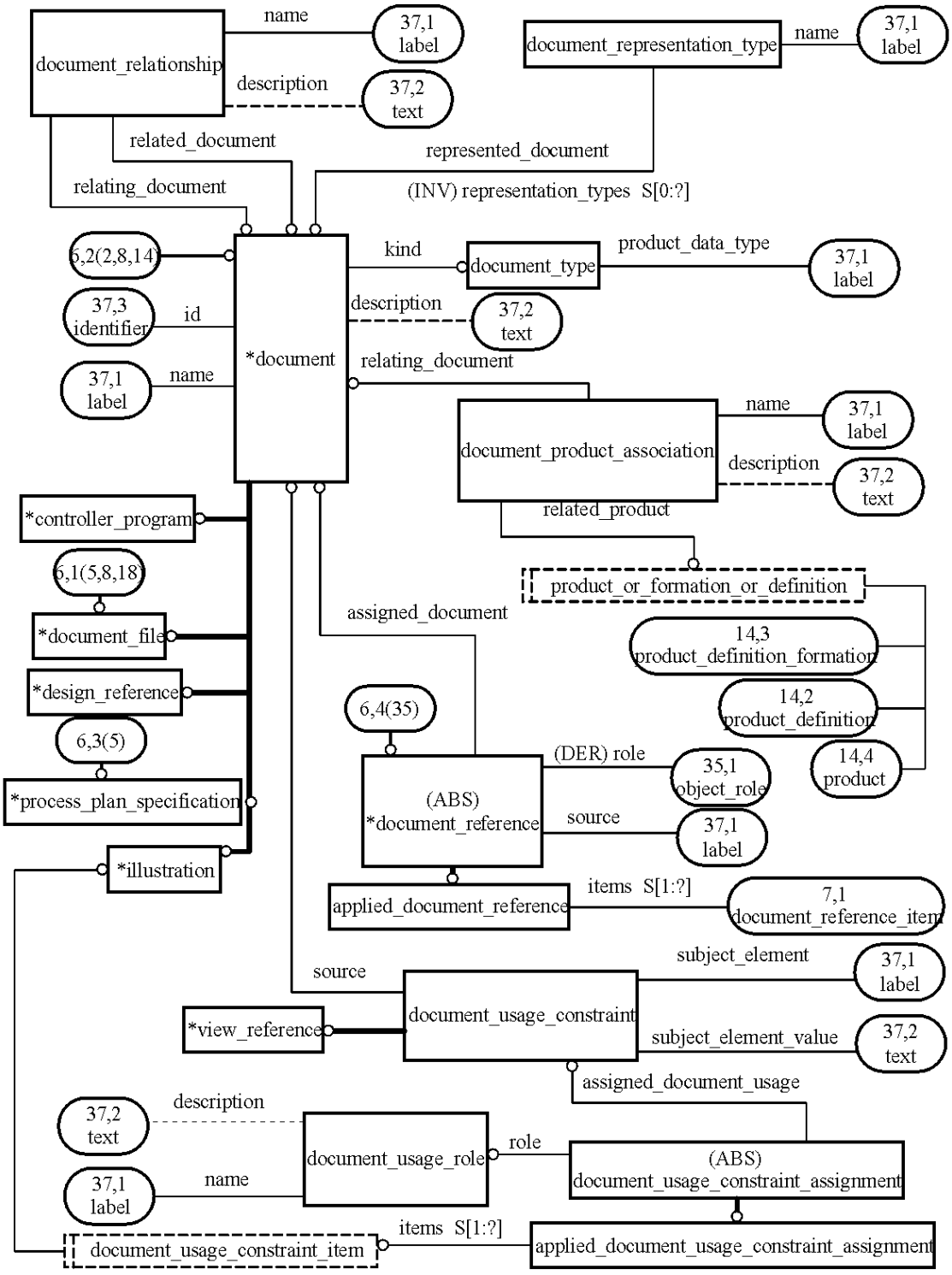


**Figure H.6 — document- AIM EXPRESS-G diagram 6 of 36**

 

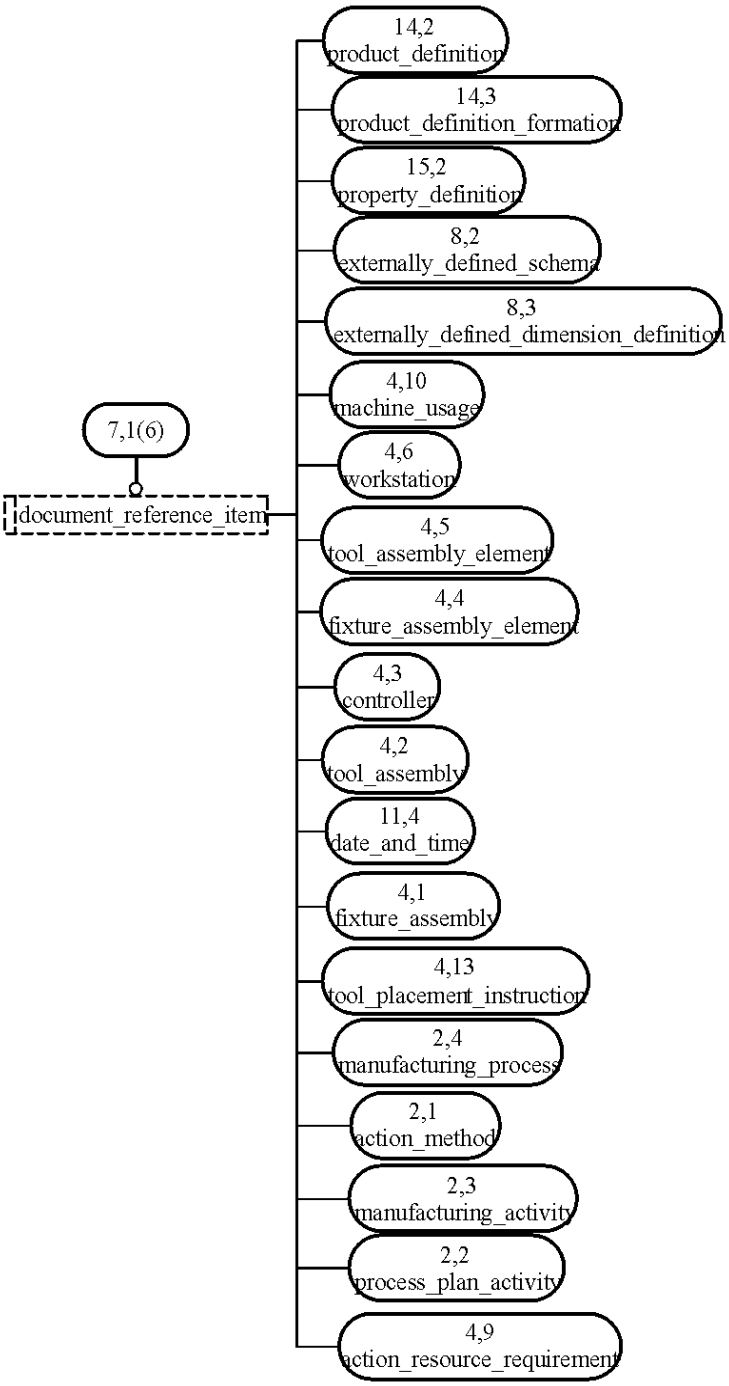


**Figure H.7 — document reference - AIM EXPRESS-G diagram 7 of 36**

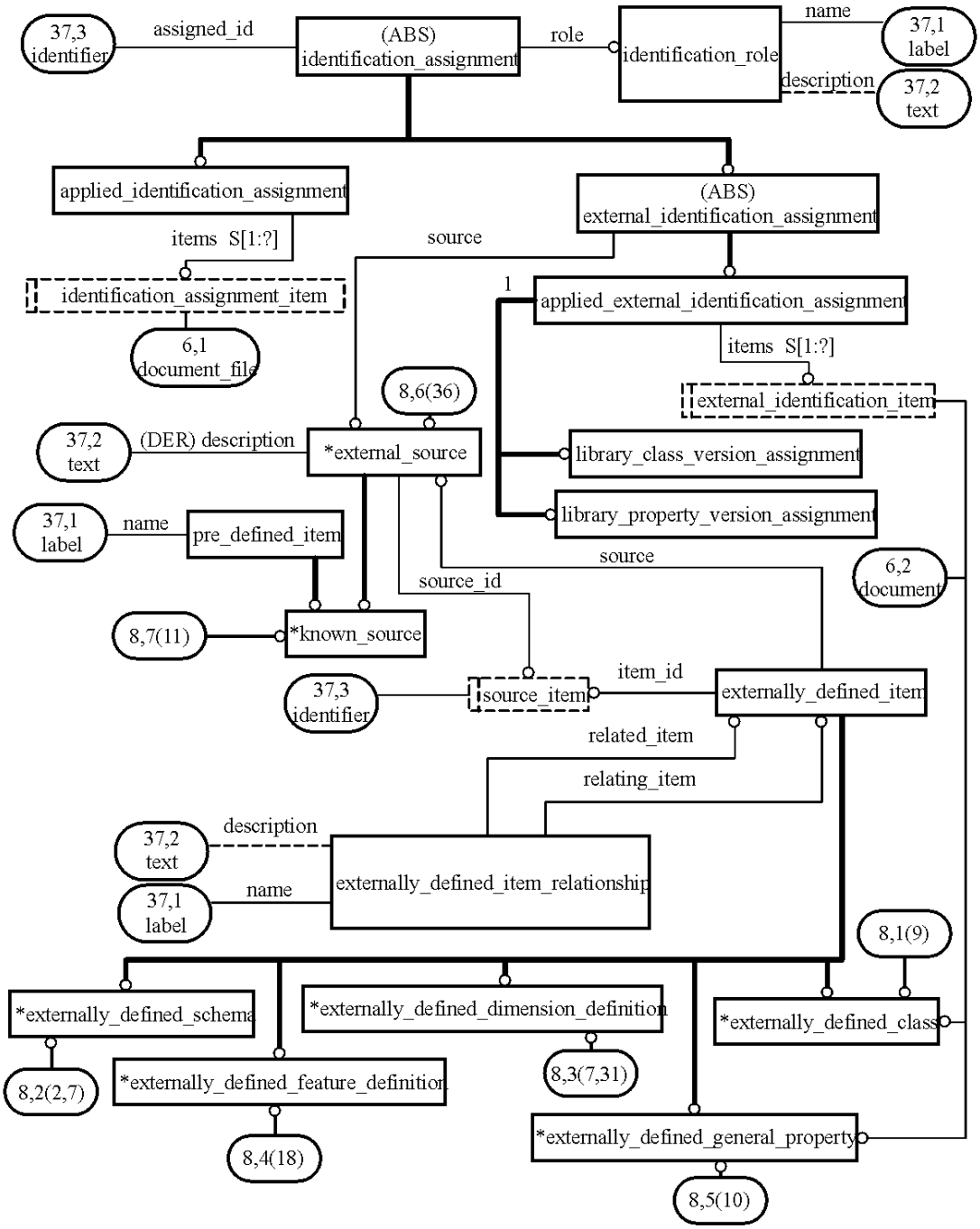


**Figure H.8 — identification assignment - AIM EXPRESS-G diagram 8 of 36**

 

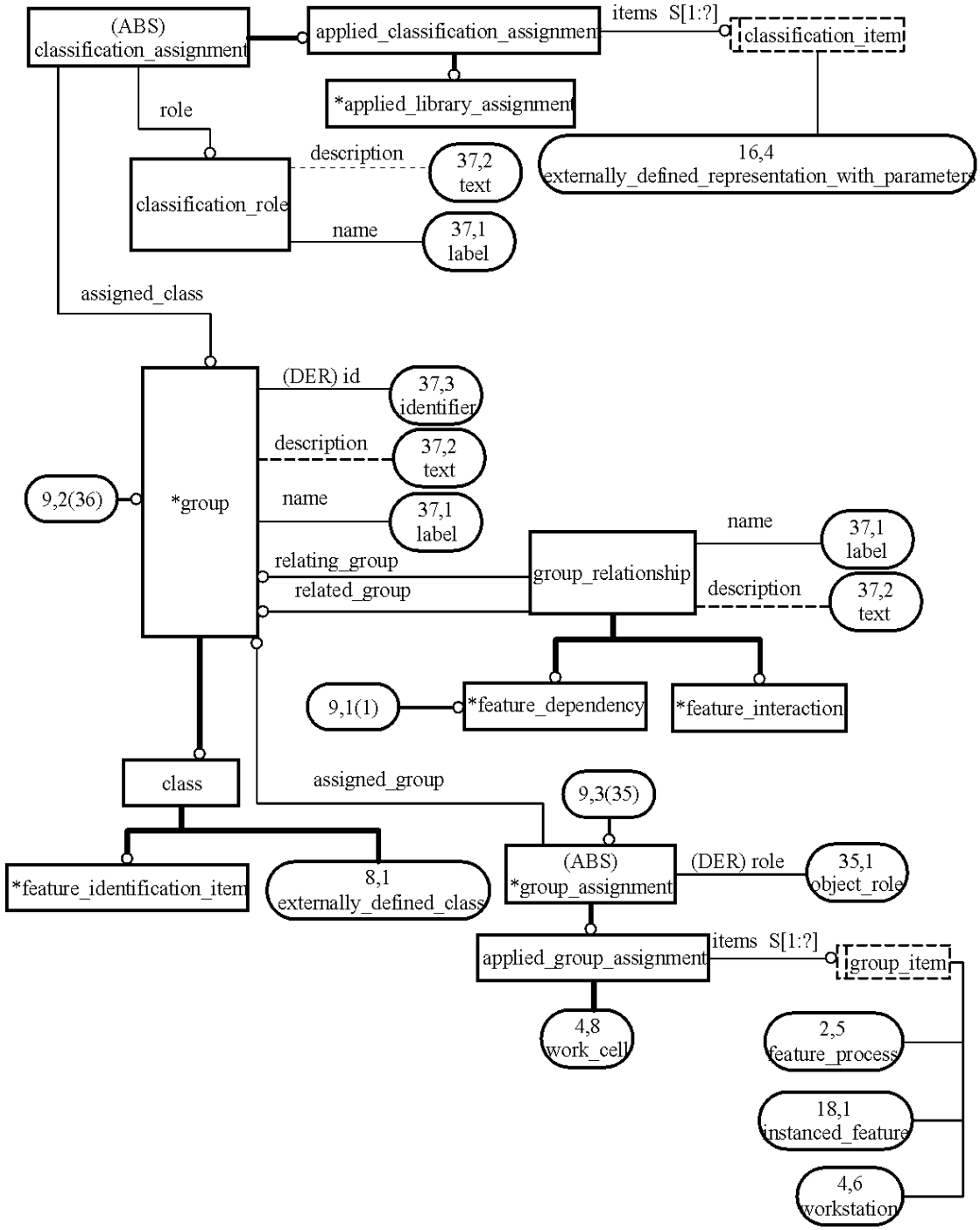


**Figure H.9 — group - AIM EXPRESS-G diagram 9 of 36**

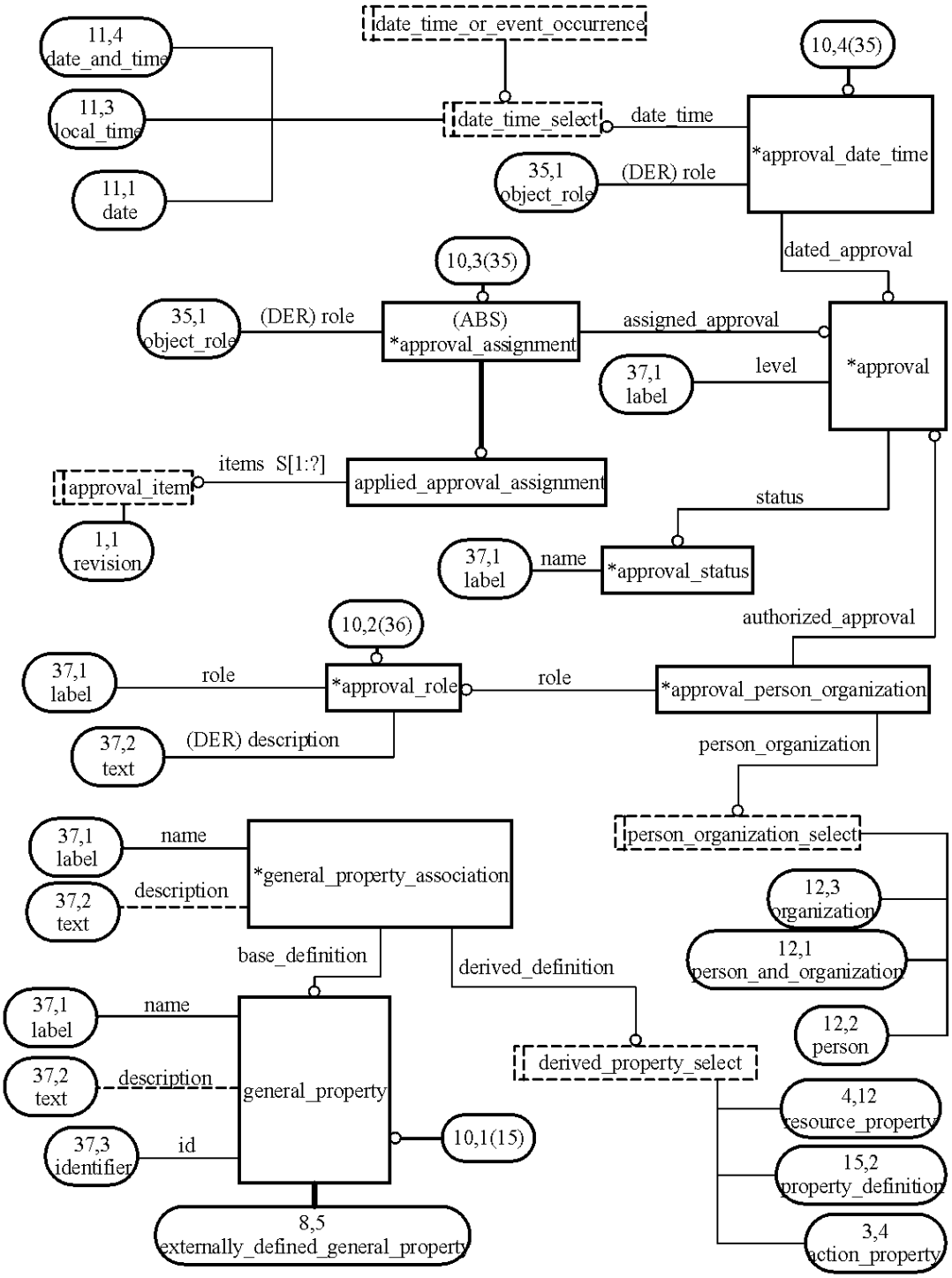


**Figure H.10 — approval- AIM EXPRESS-G diagram 10 of 36**

 

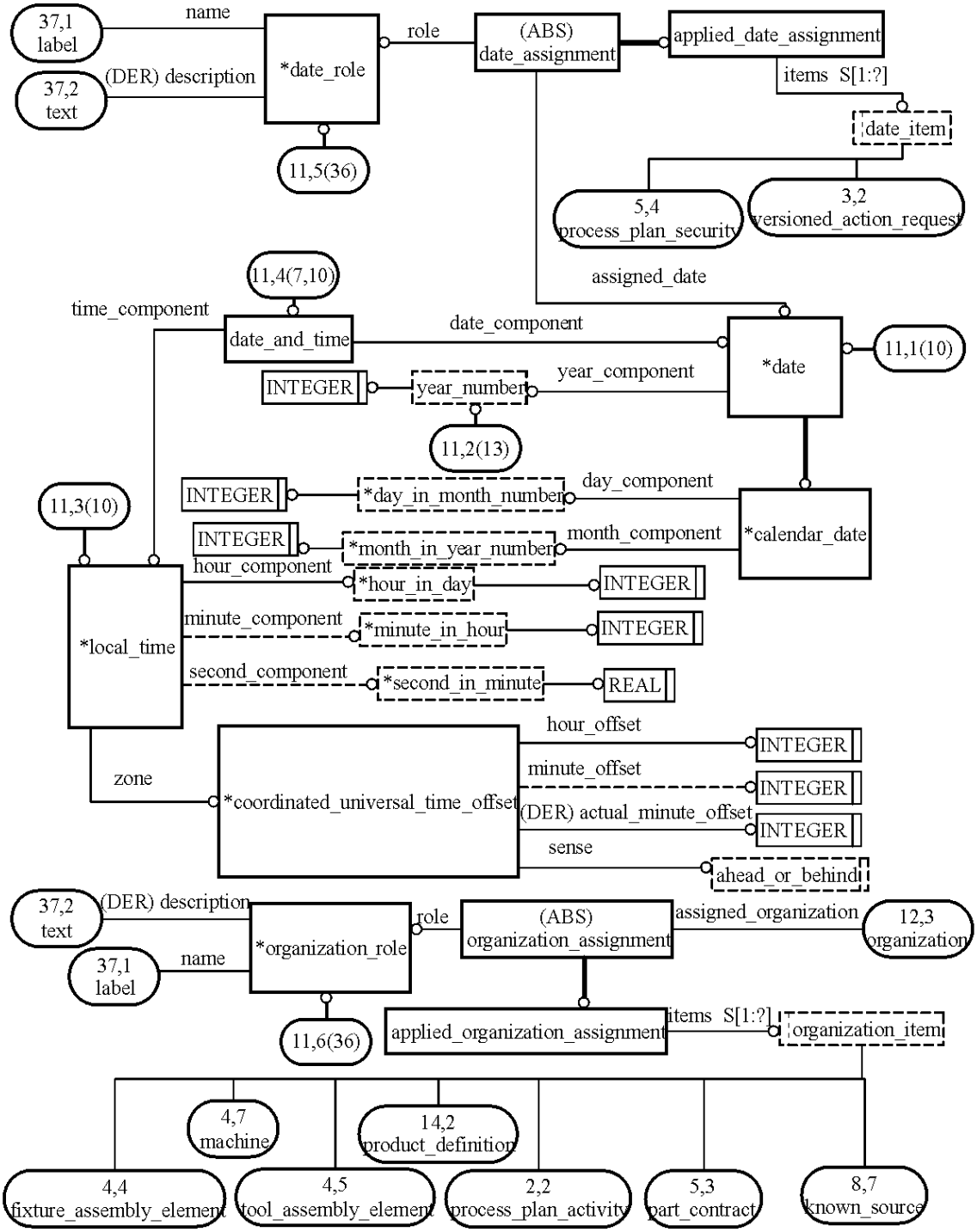


**Figure H.11 — date - AIM EXPRESS-G diagram 11 of 36**

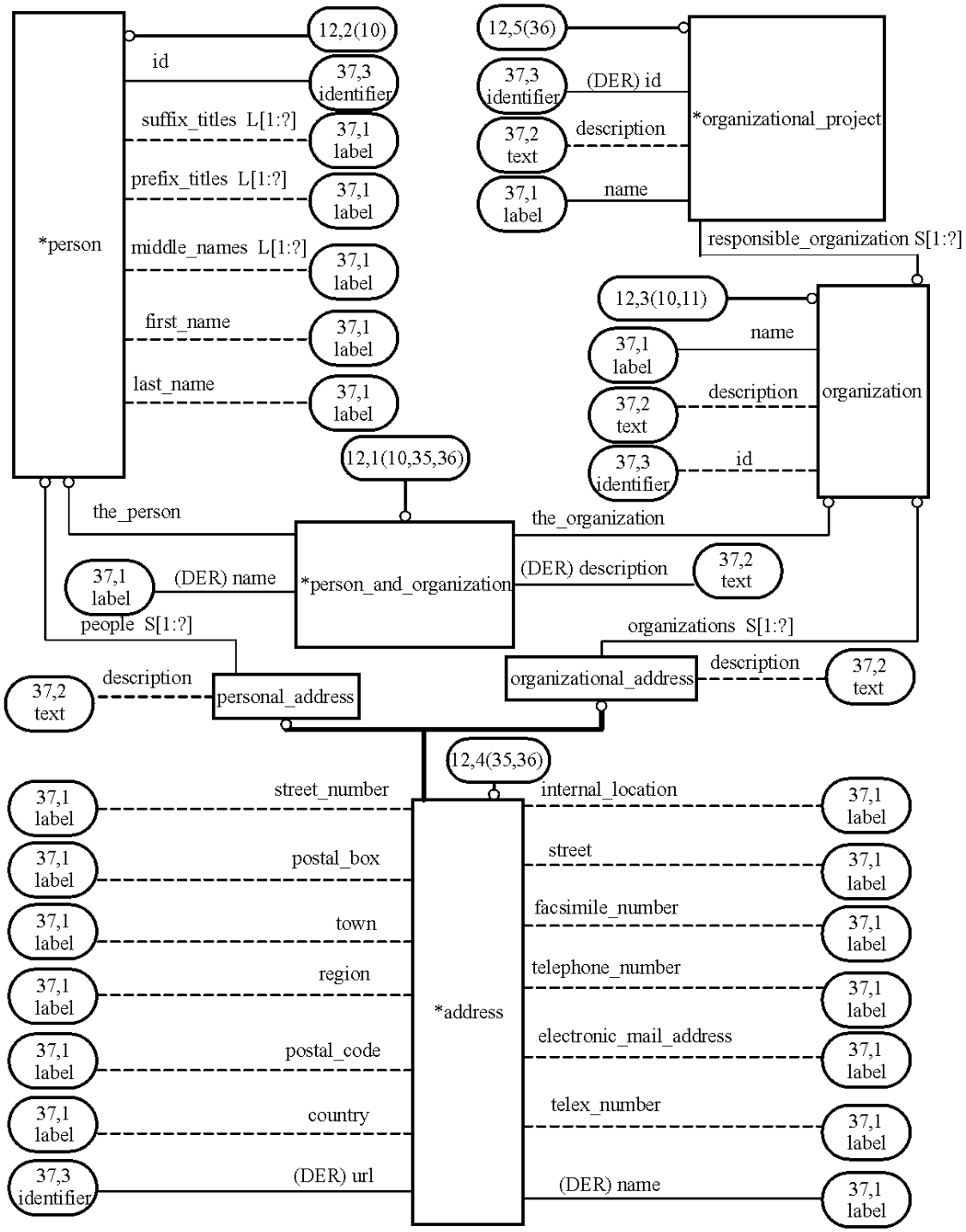


**Figure H.12 — person and organization - AIM EXPRESS-G diagram 12 of 36**

 

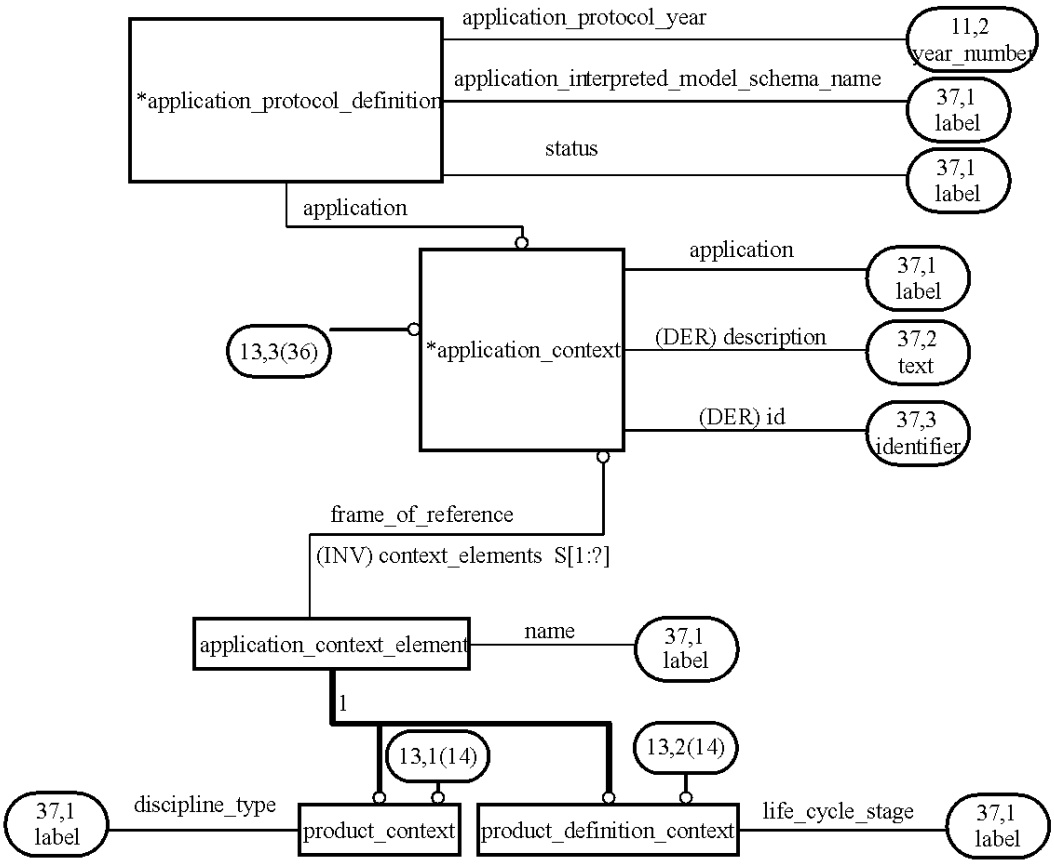


**Figure H.13 — application context - AIM EXPRESS-G diagram 13 of 36**

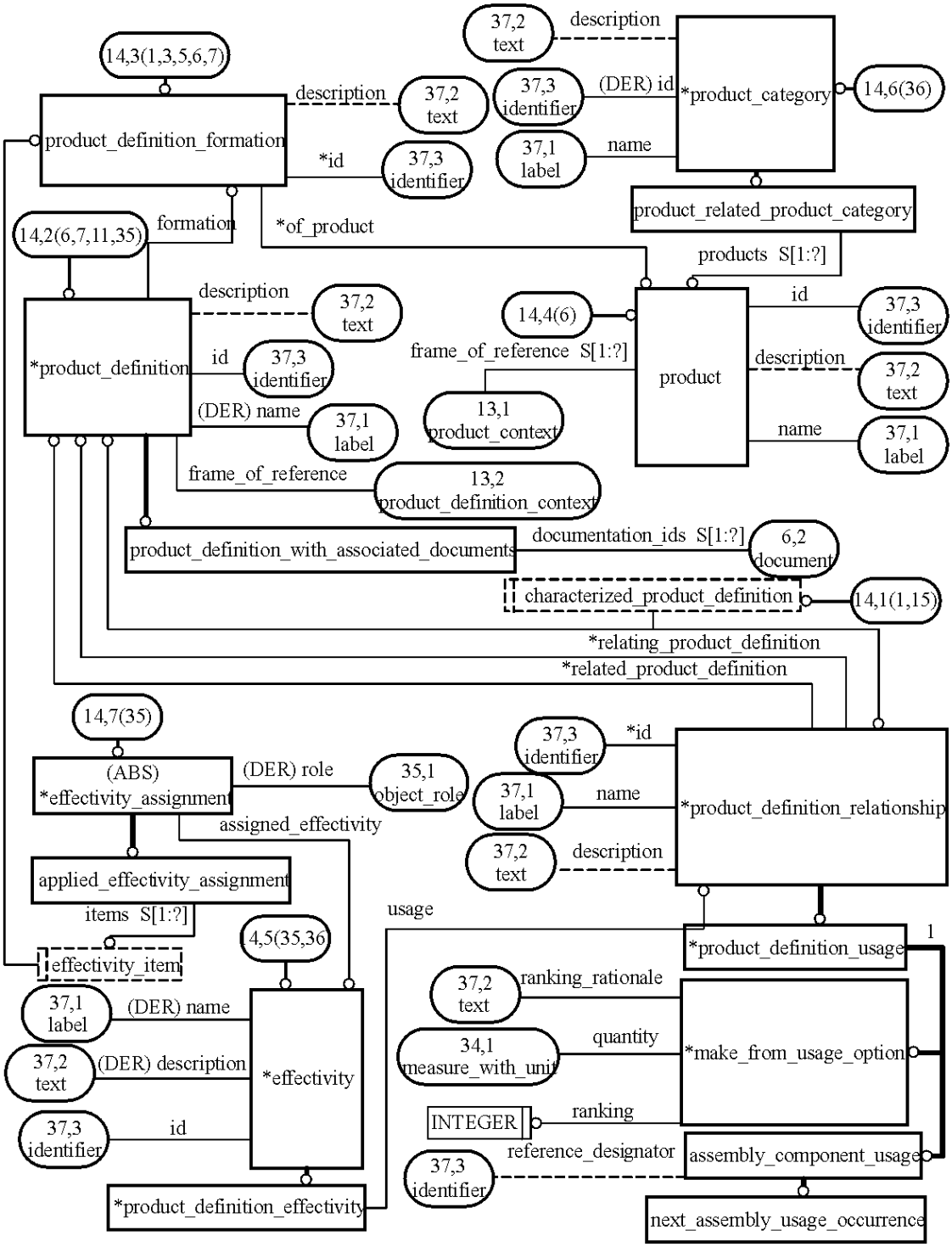


**Figure H.14 — product - AIM EXPRESS-G diagram 14 of 36**

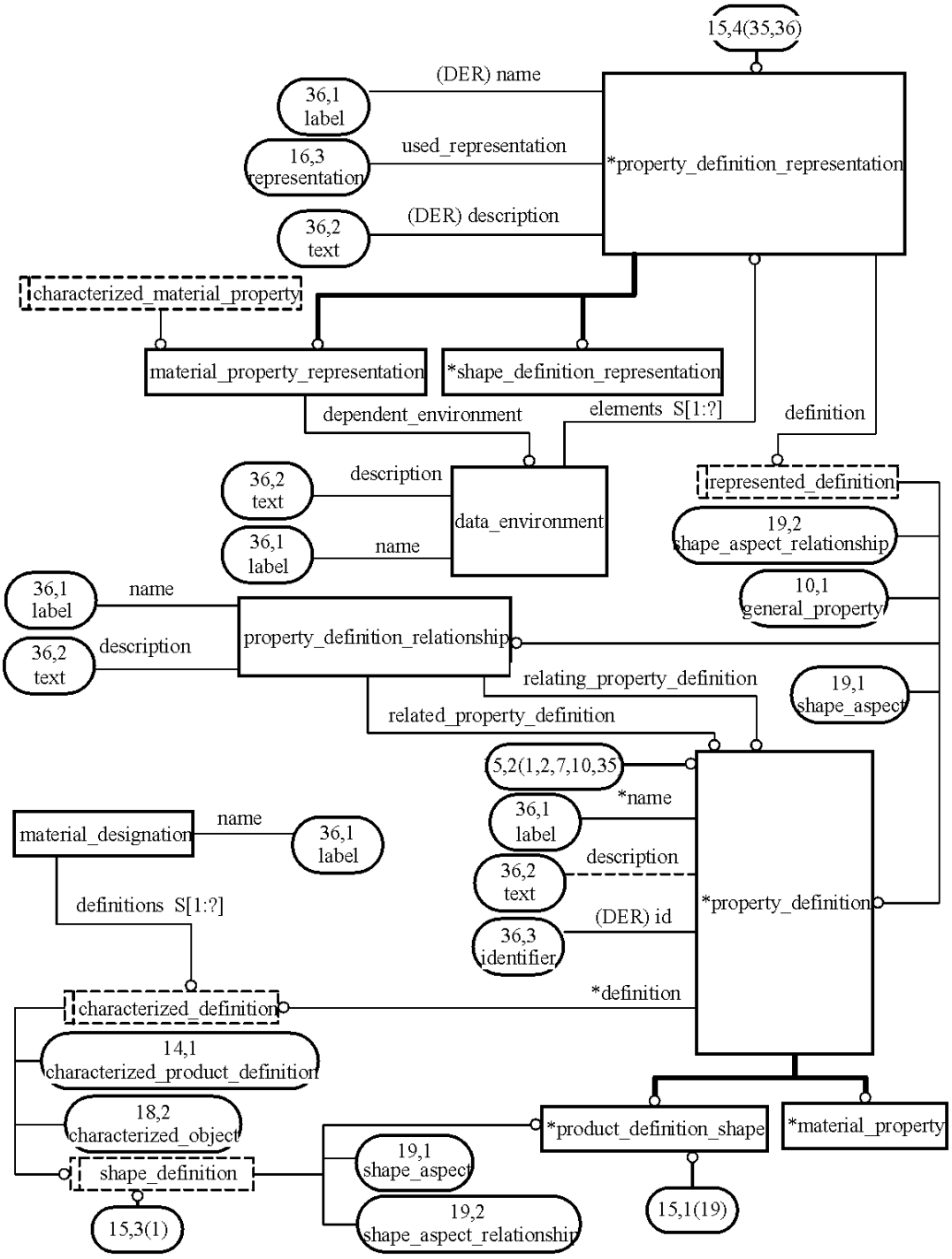


**Figure H.15 — property definition - AIM EXPRESS-G diagram 15 of 36**

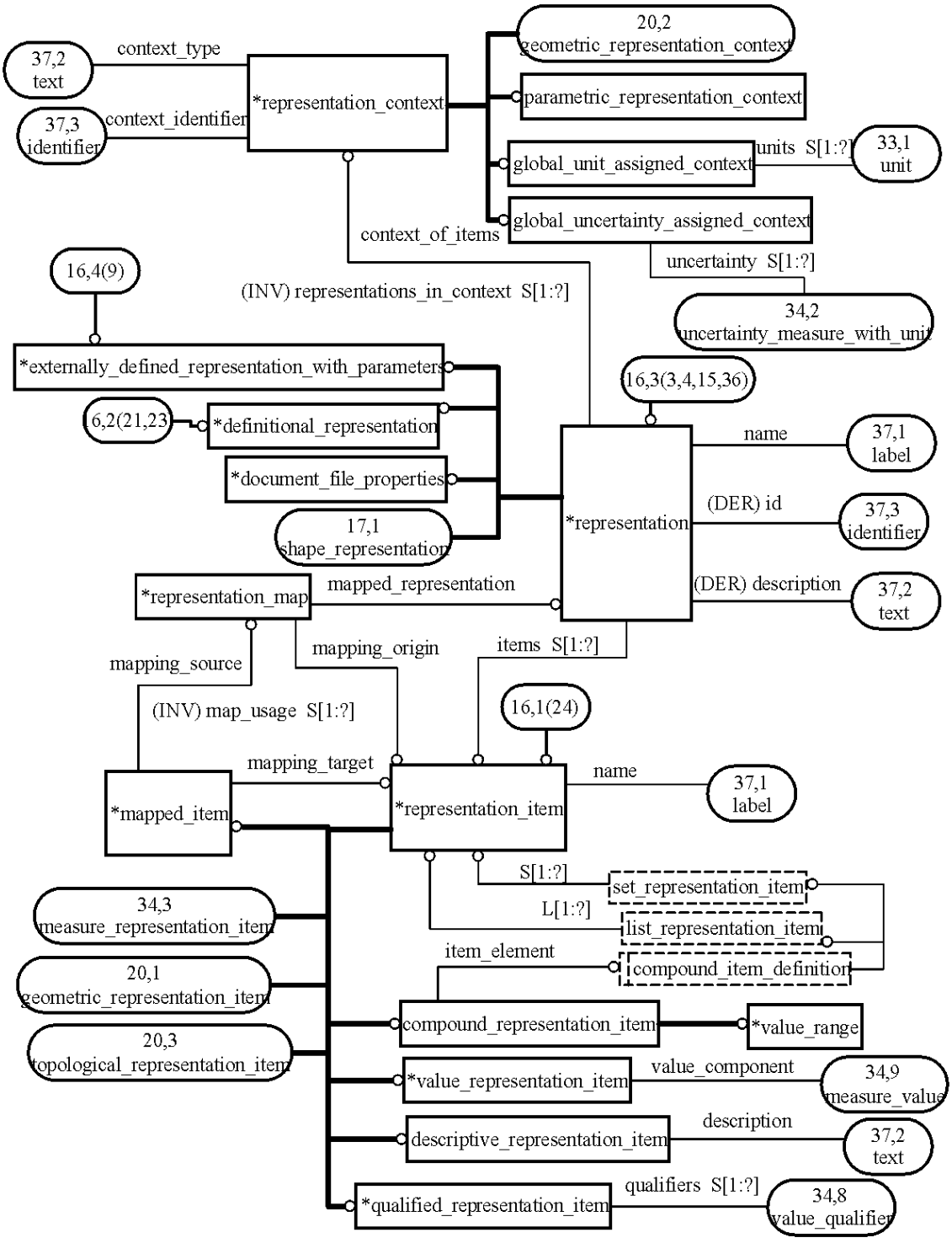


**Figure H.16 — representation - AIM EXPRESS-G diagram 16 of 36**

 

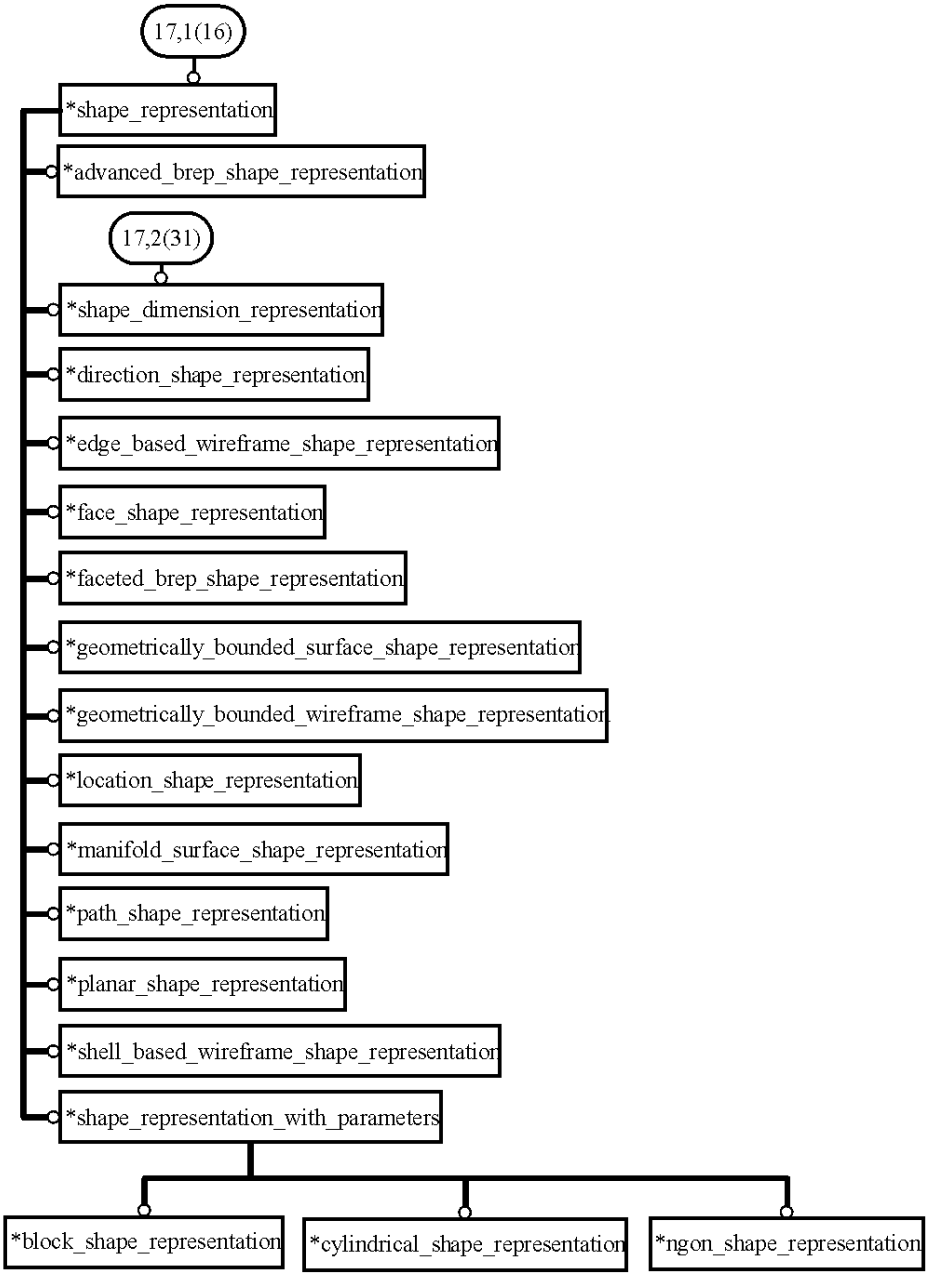


**Figure H.17 — shape representation - AIM EXPRESS-G diagram 17 of 36**

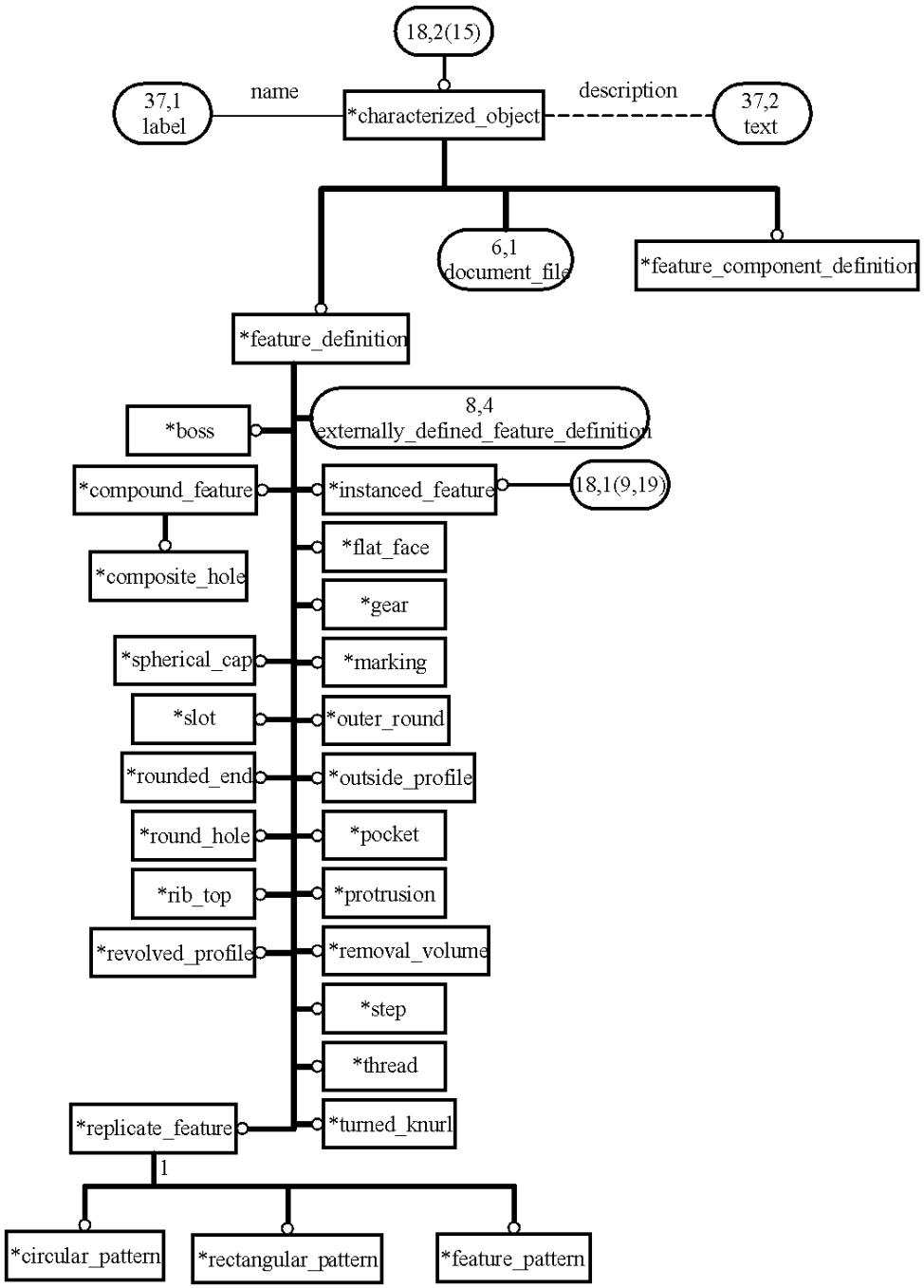


**Figure H.18 — characterized object - AIM EXPRESS-G diagram 18 of 36**

 

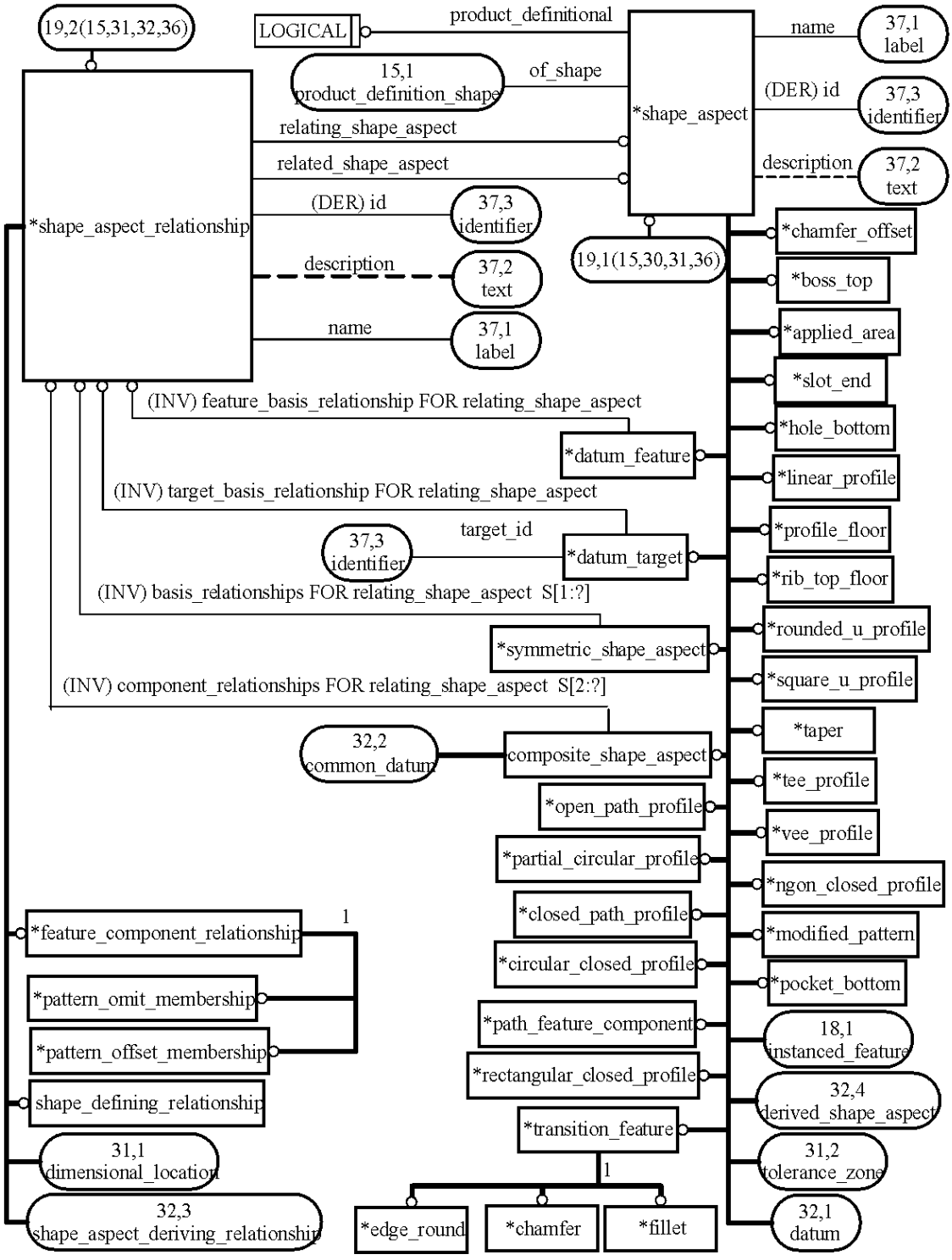


**Figure H.19 — shape aspect- AIM EXPRESS-G diagram 19 of 36**

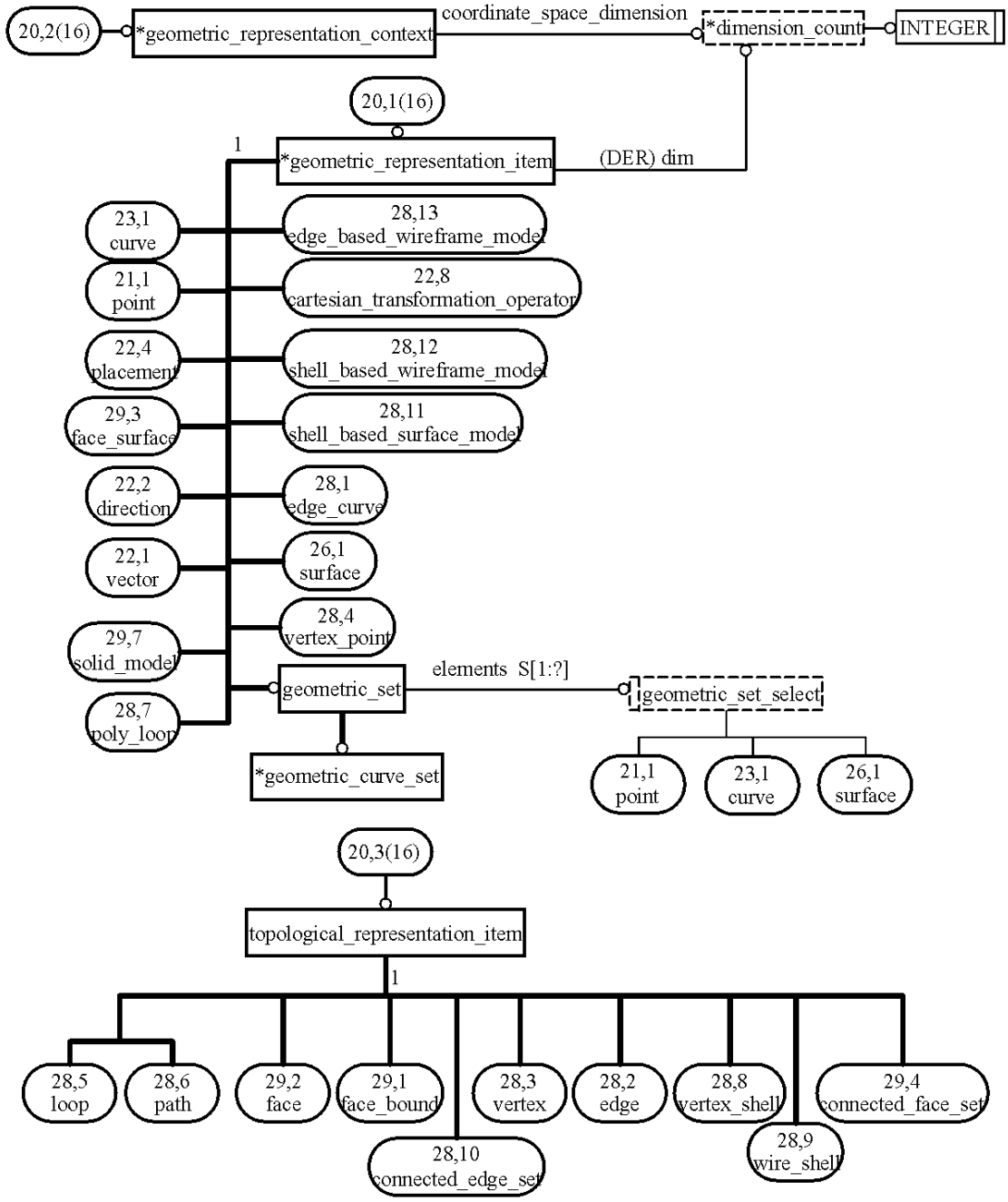


**Figure H.20 — geometry topology - AIM EXPRESS-G diagram 20 of 36**

 

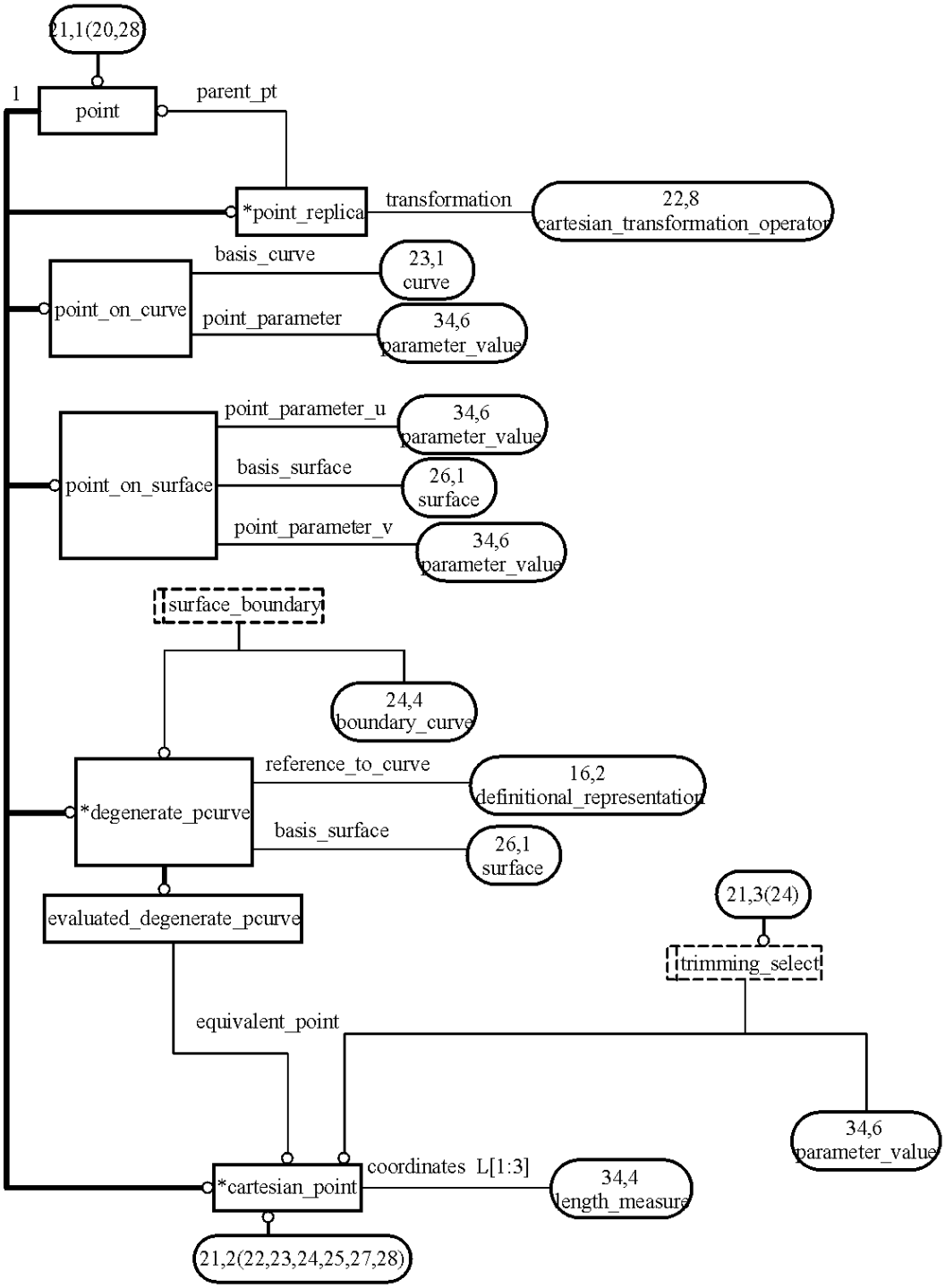


**Figure H.21 — point - AIM EXPRESS-G diagram 21 of 36**

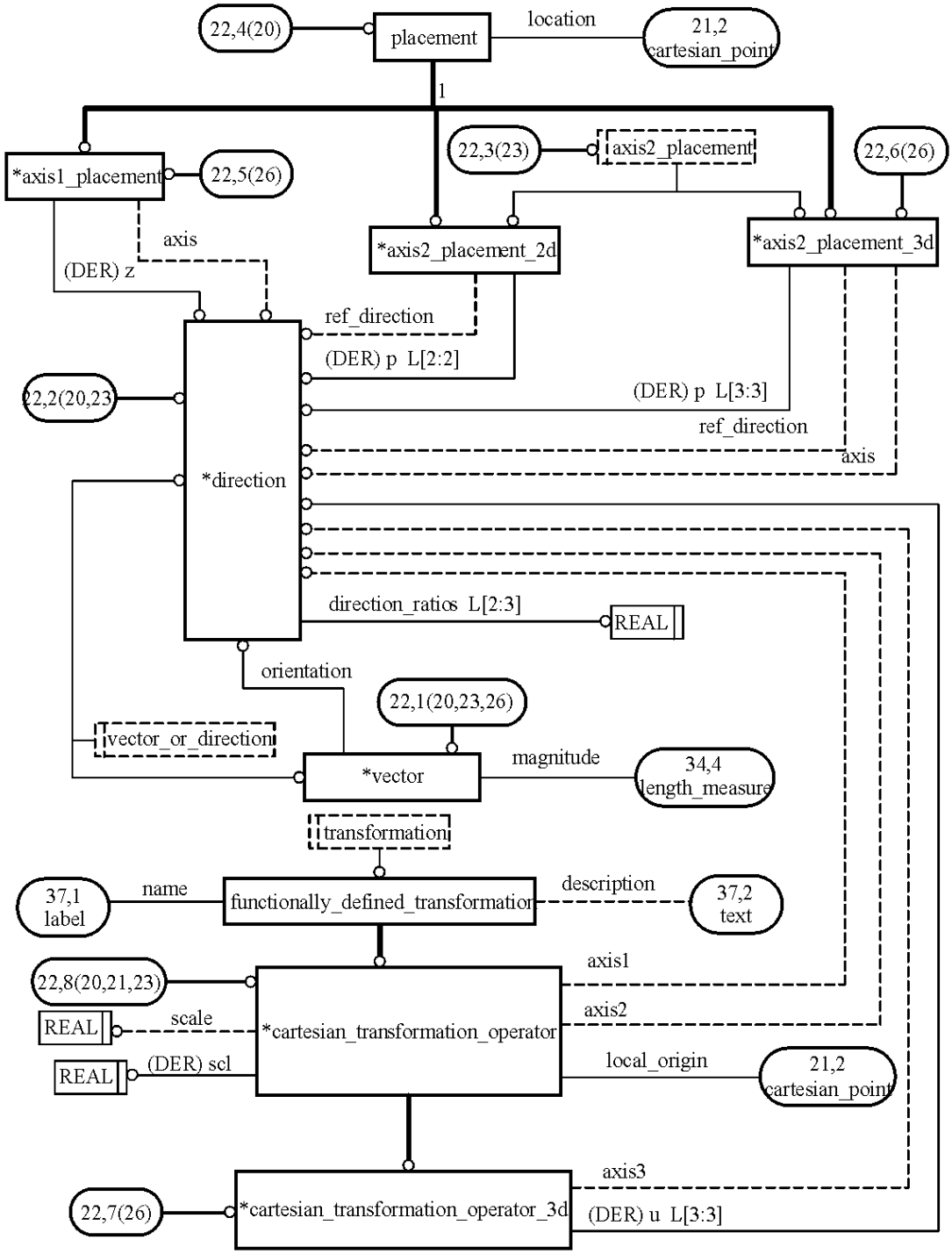


**Figure H.22 — position - AIM EXPRESS-G diagram 22 of 36**

 

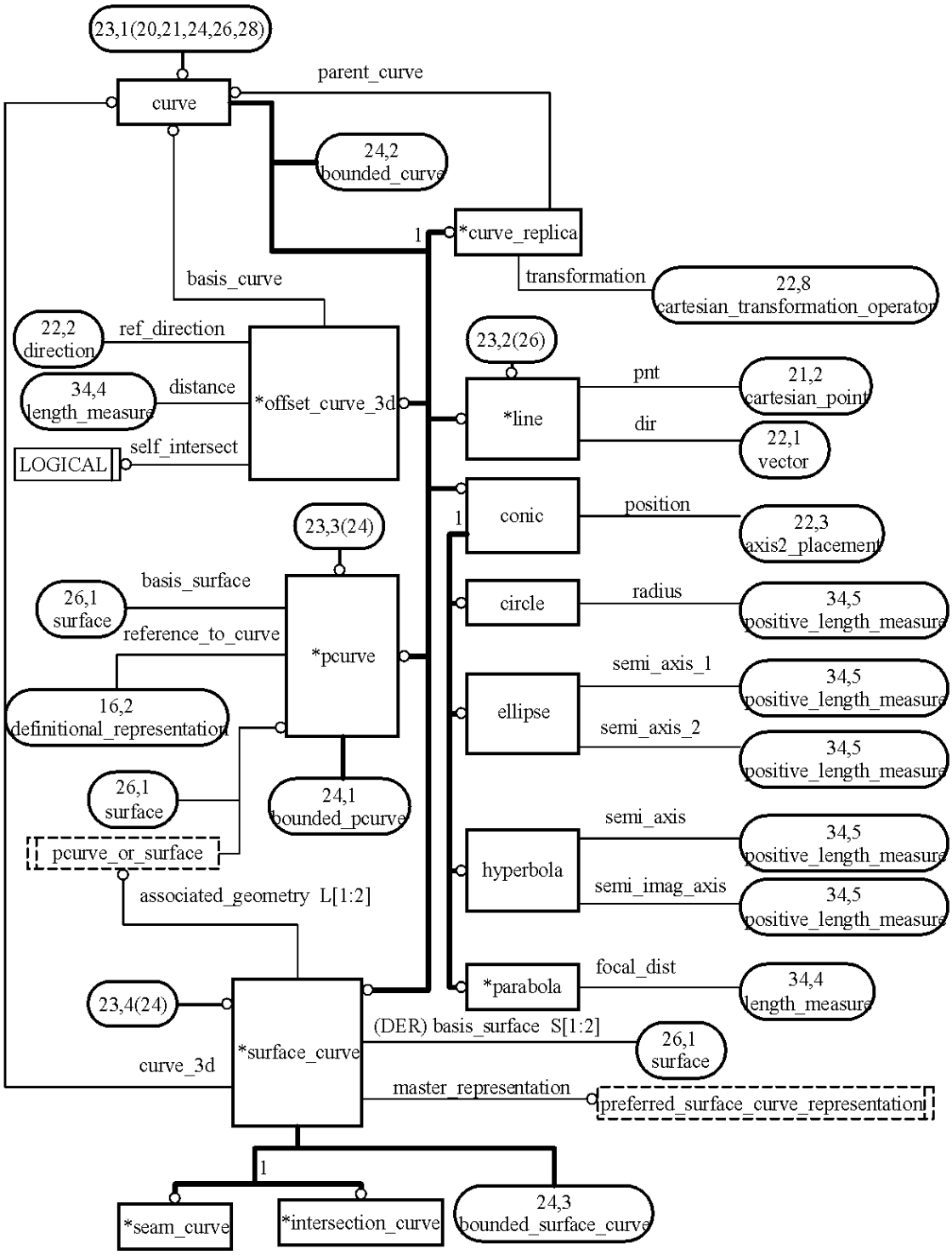


**Figure H.23 — curve - AIM EXPRESS-G diagram 23 of 36**

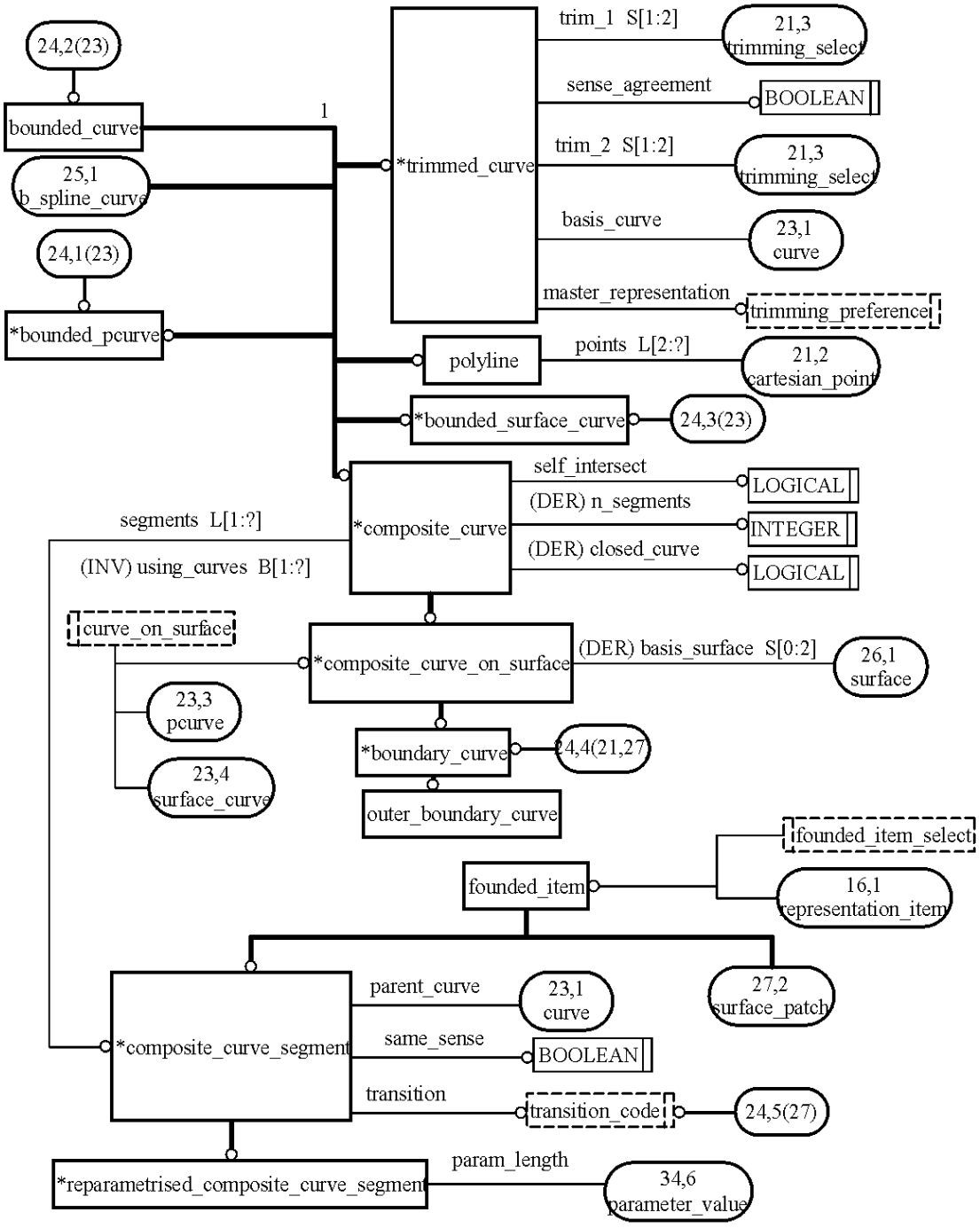


**Figure H.24 — bounded curve - AIM EXPRESS-G diagram 24 of 36**

 

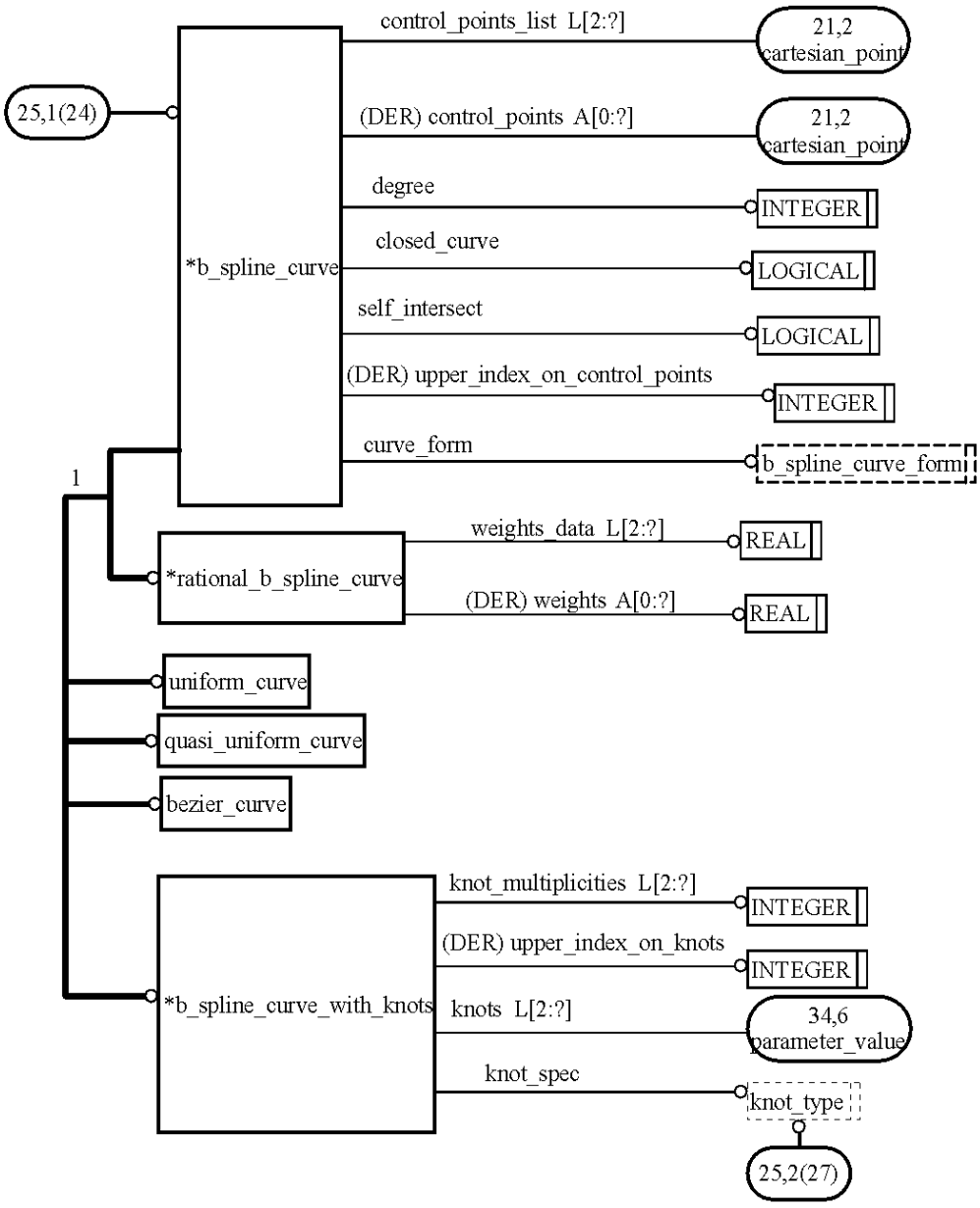


**Figure H.25 — b_spline curve - AIM EXPRESS-G diagram 25 of 36**

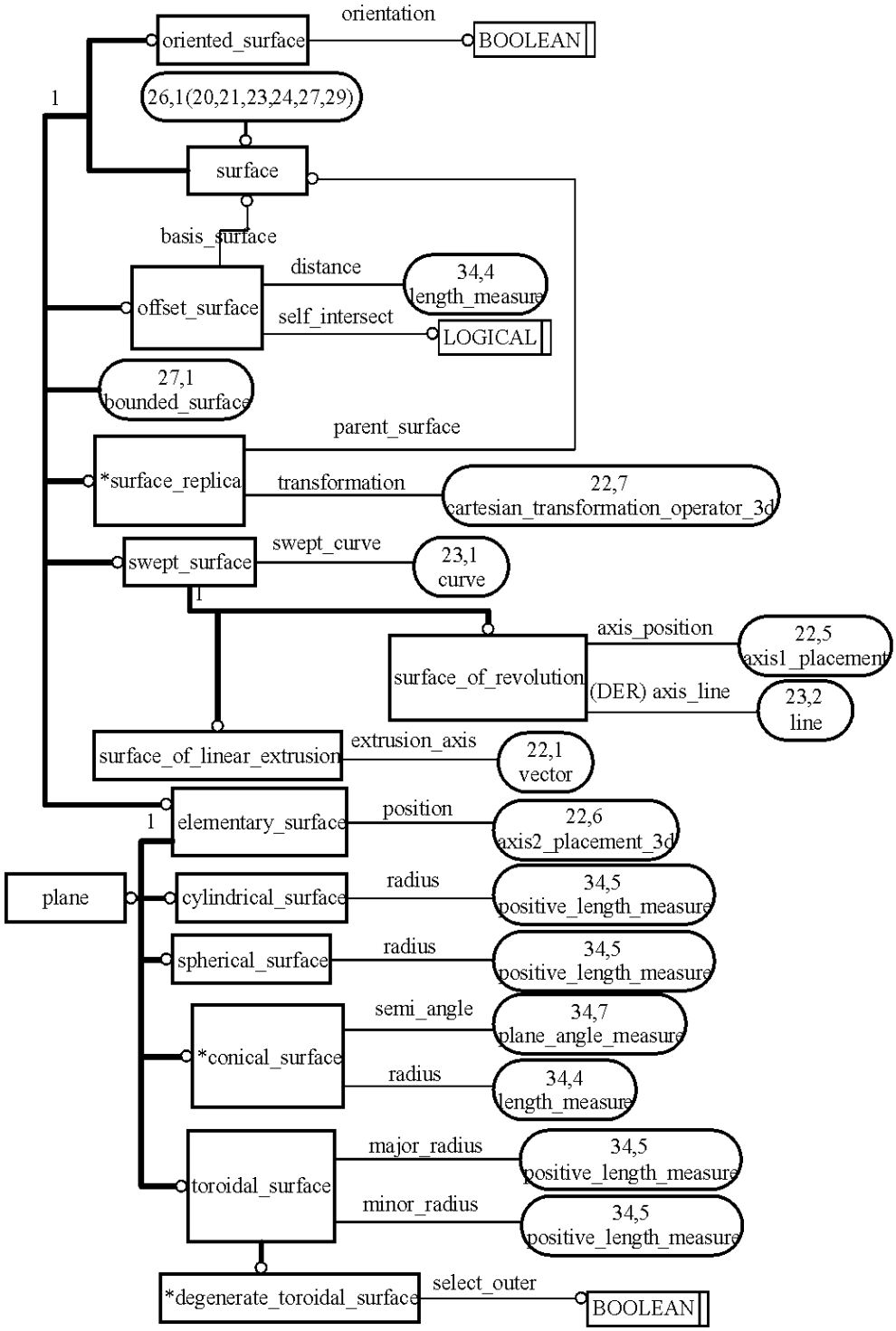


**Figure H.26 — surface - AIM EXPRESS-G diagram 26 of 36**

 

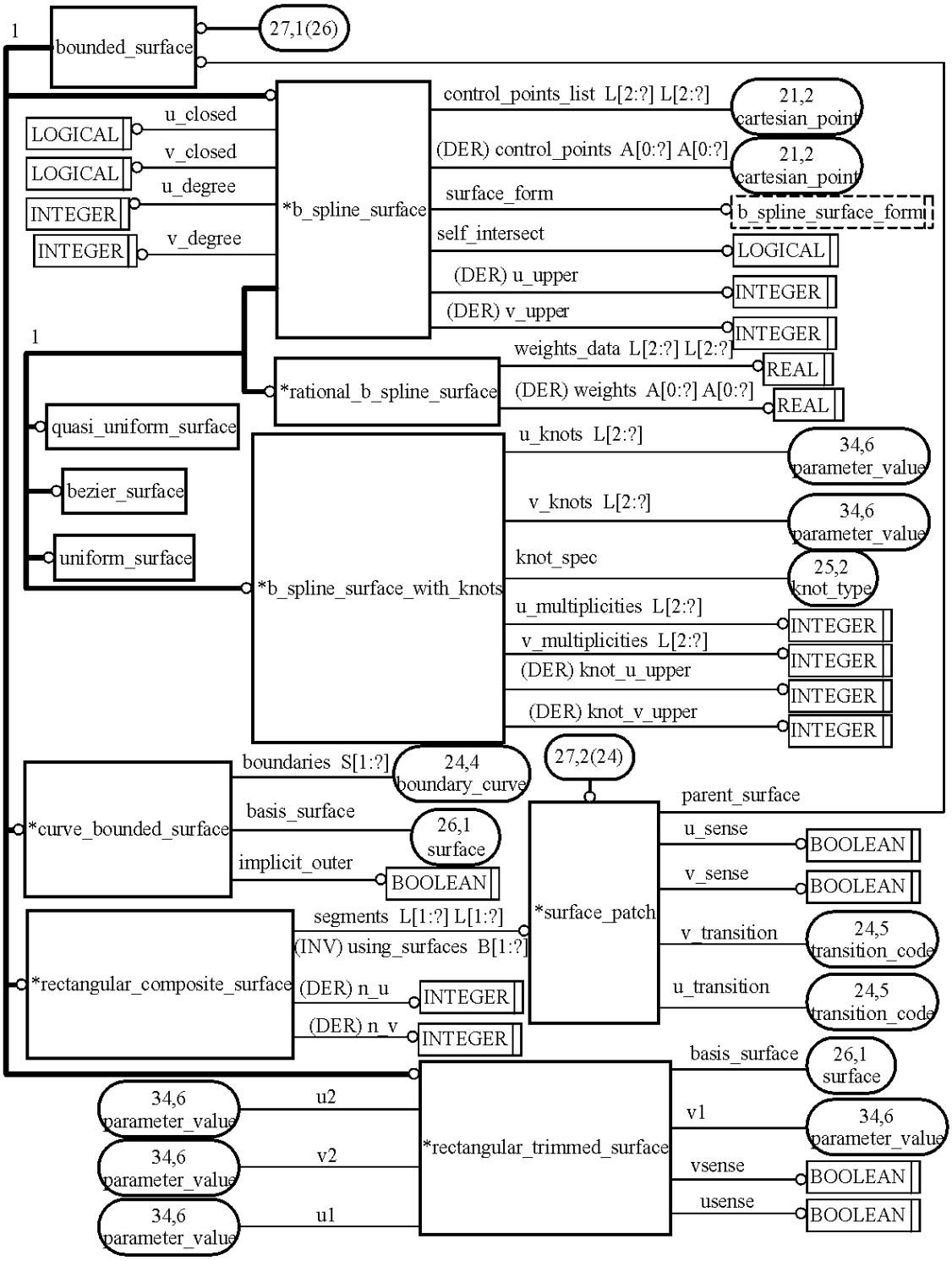


**Figure H.27 — bounded surface - AIM EXPRESS-G diagram 27 of 36**

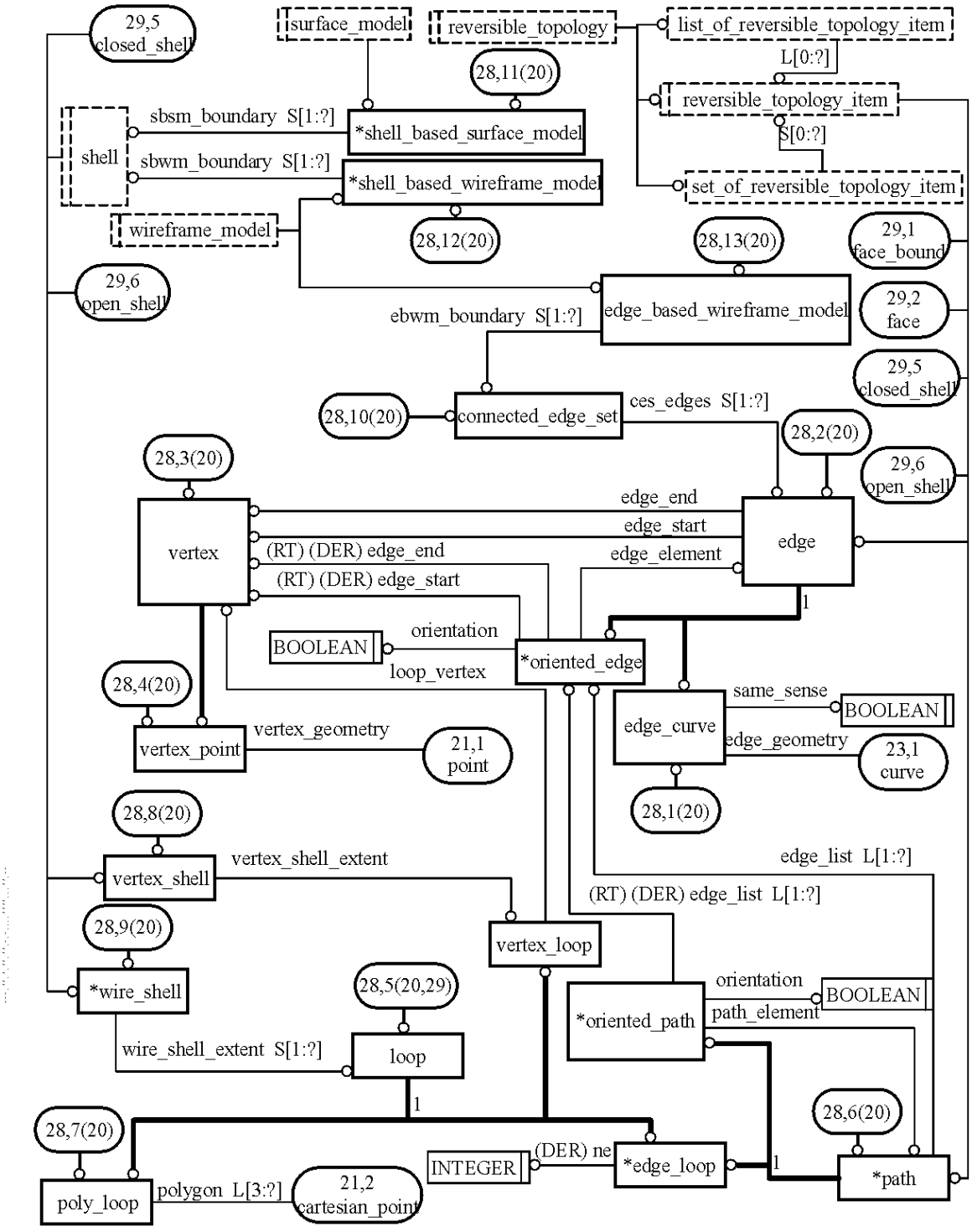


**Figure H.28 — topology - AIM EXPRESS-G diagram 28 of 36**

 

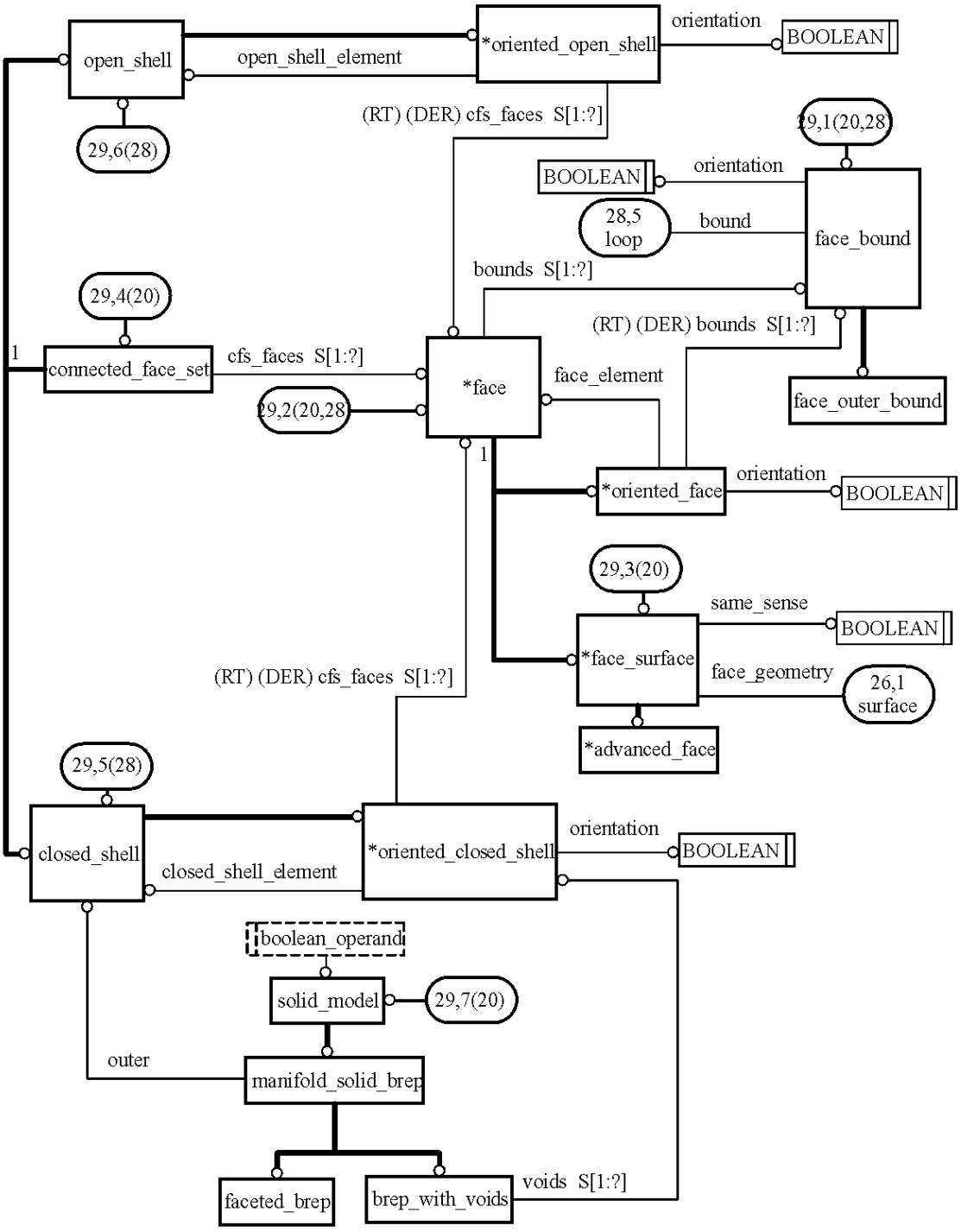


**Figure H.29 — face and shell - AIM EXPRESS-G diagram 29 of 36**

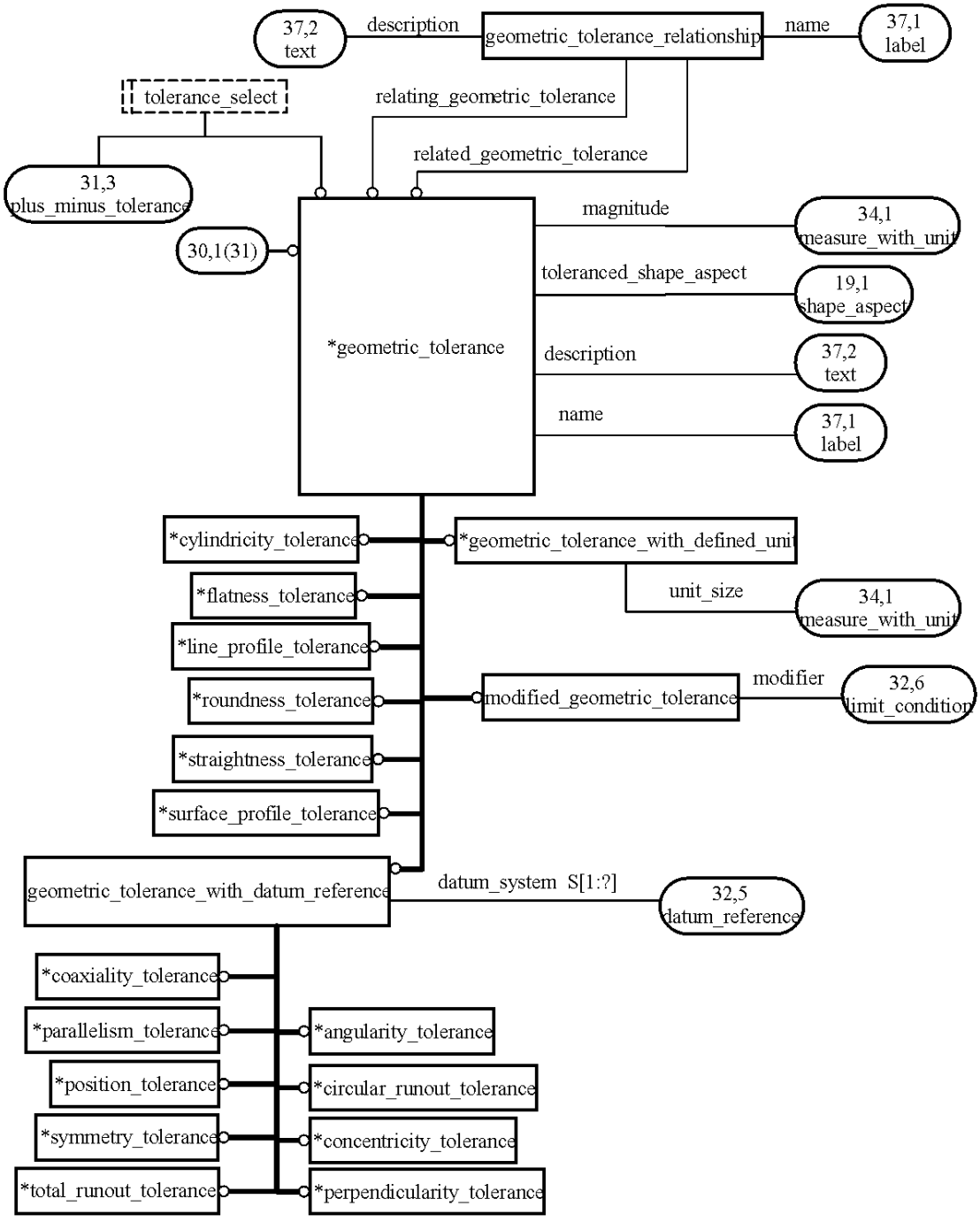


**Figure H.30 — geometric tolerance - AIM EXPRESS-G diagram 30 of 36**

 

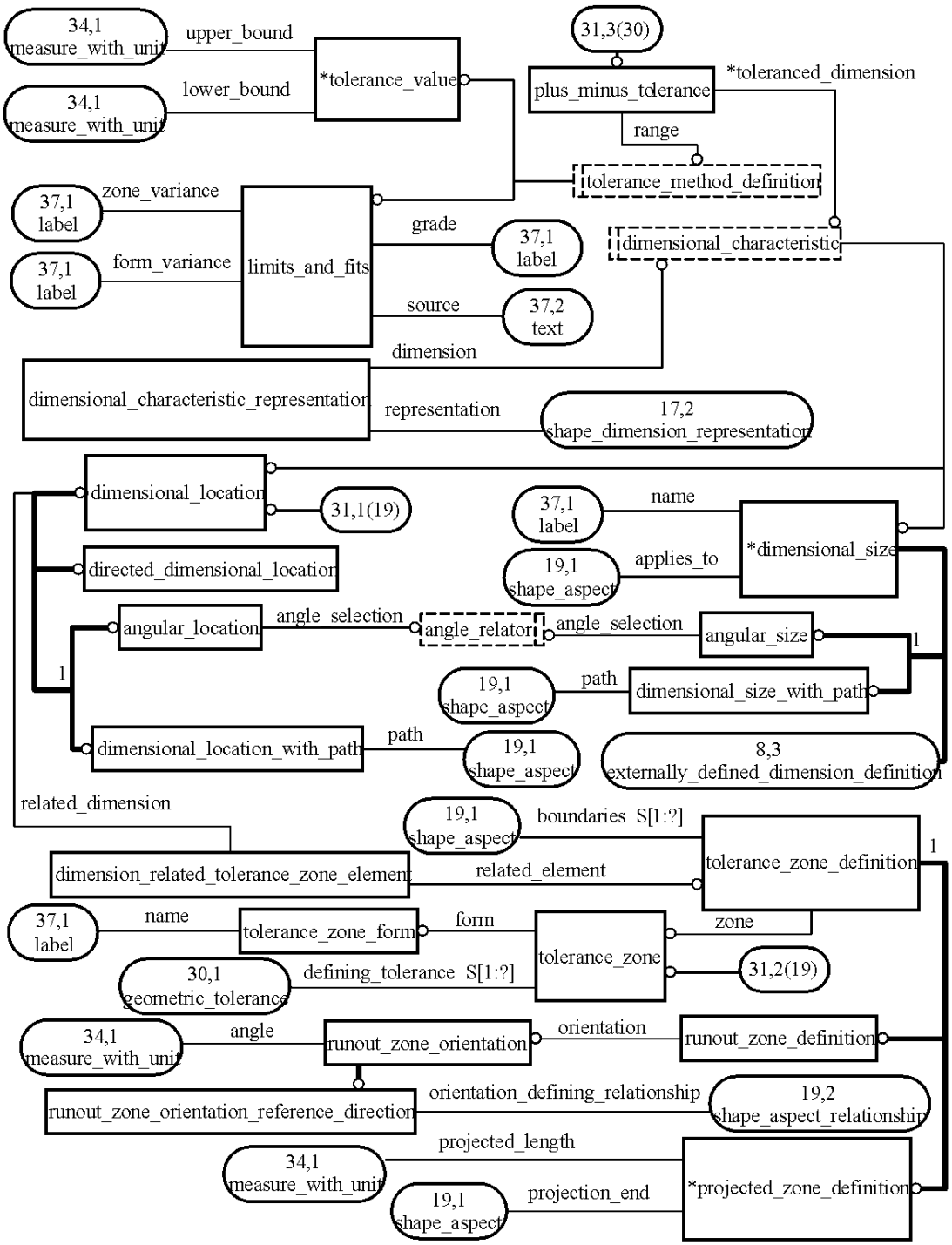


**Figure H.31 — dimensional tolerance - AIM EXPRESS-G diagram 31 of 36**

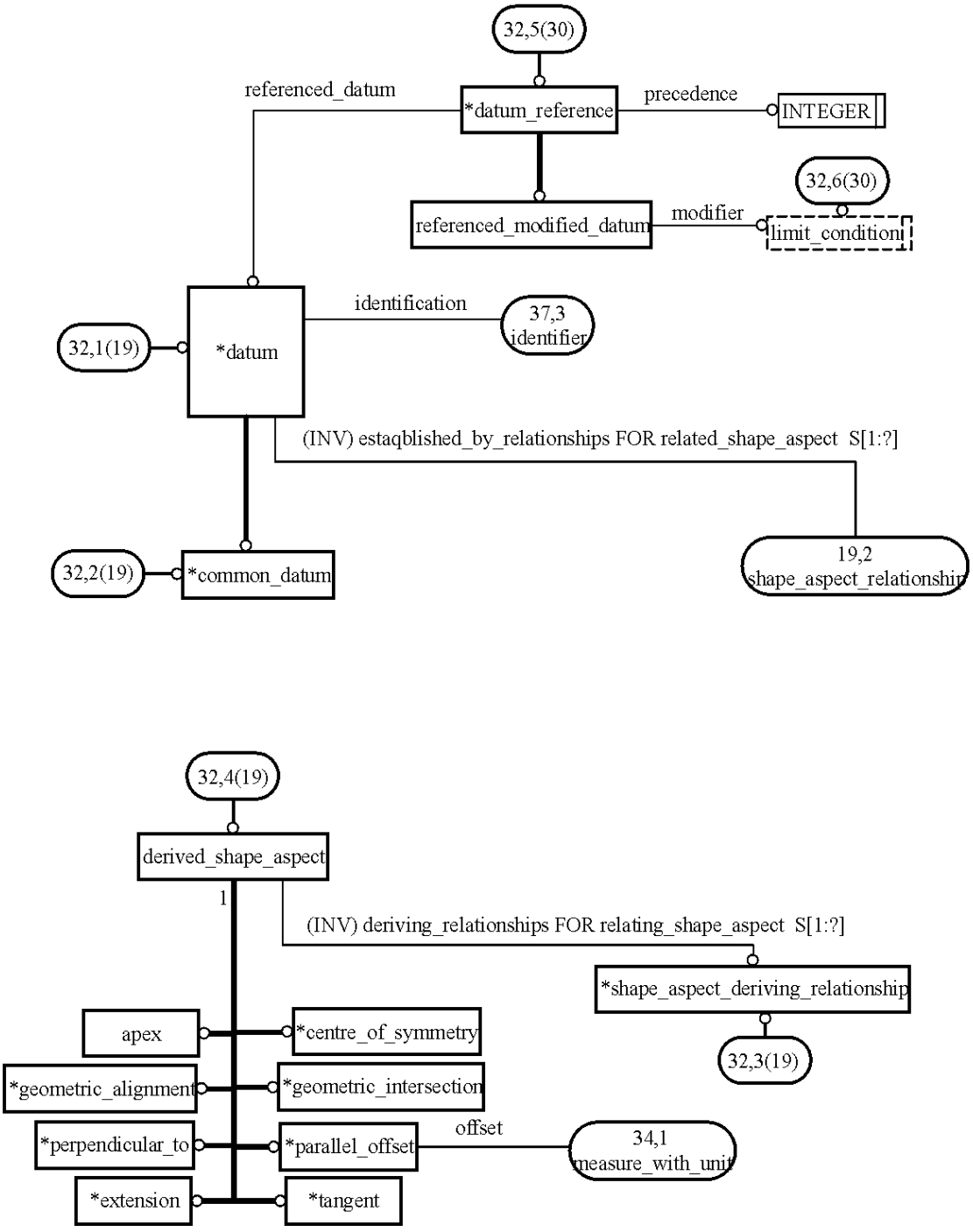


**Figure H.32 — datum - AIM EXPRESS-G diagram 32 of 36**

 

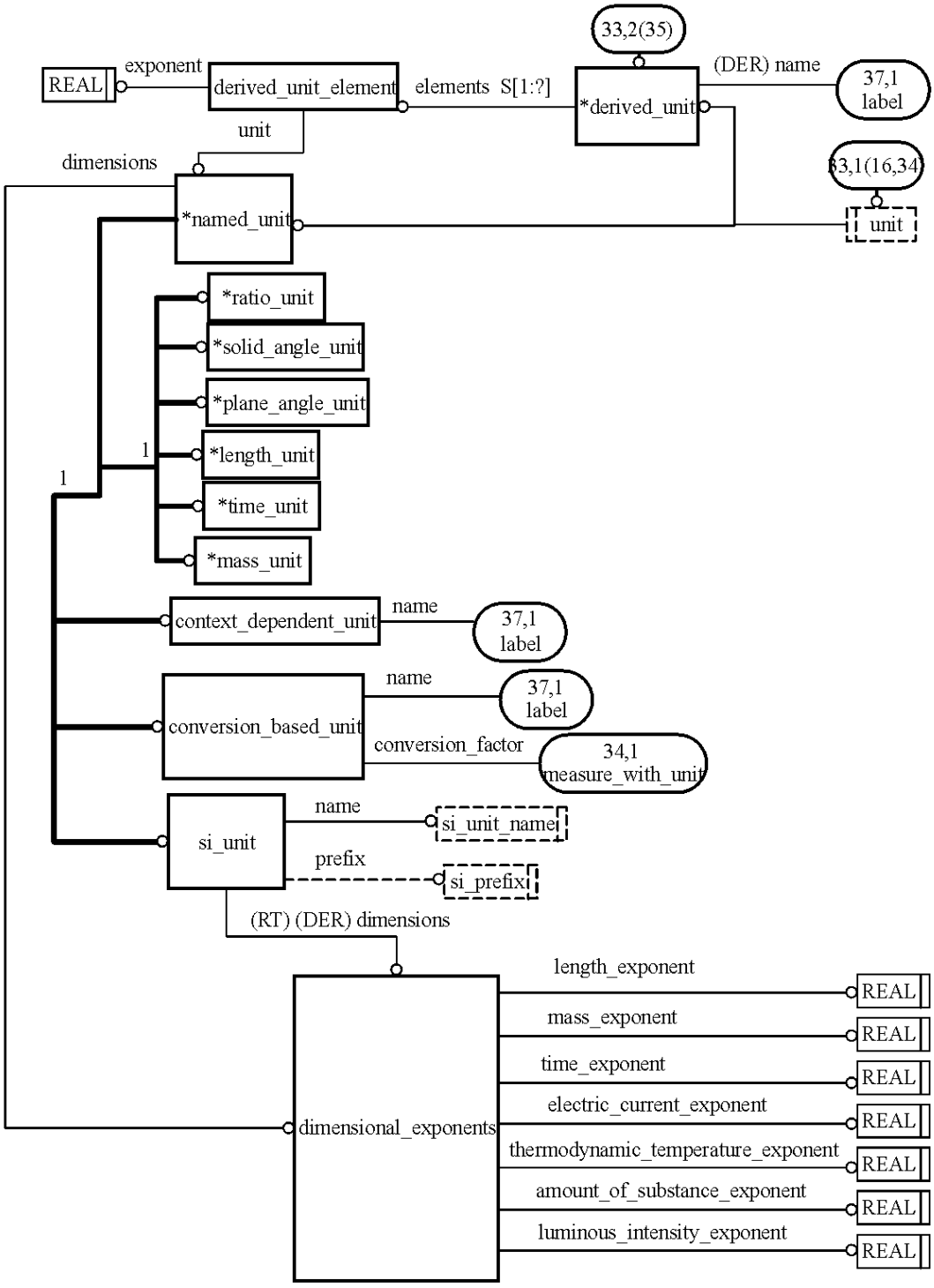


**Figure H.33 — unit - AIM EXPRESS-G diagram 33 of 36**

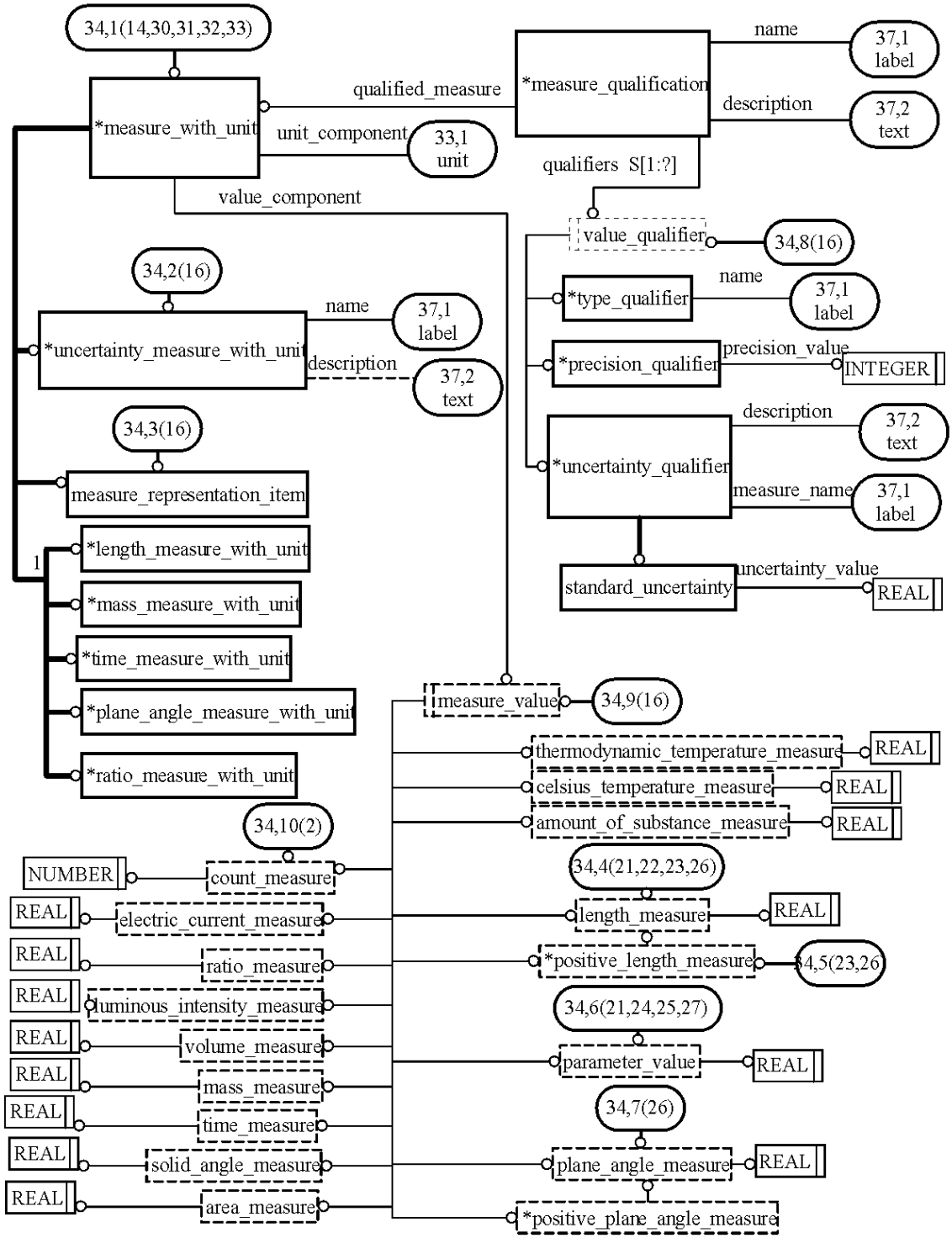


**Figure H.34 — measure with unit - AIM EXPRESS-G diagram 34 of 36**

 

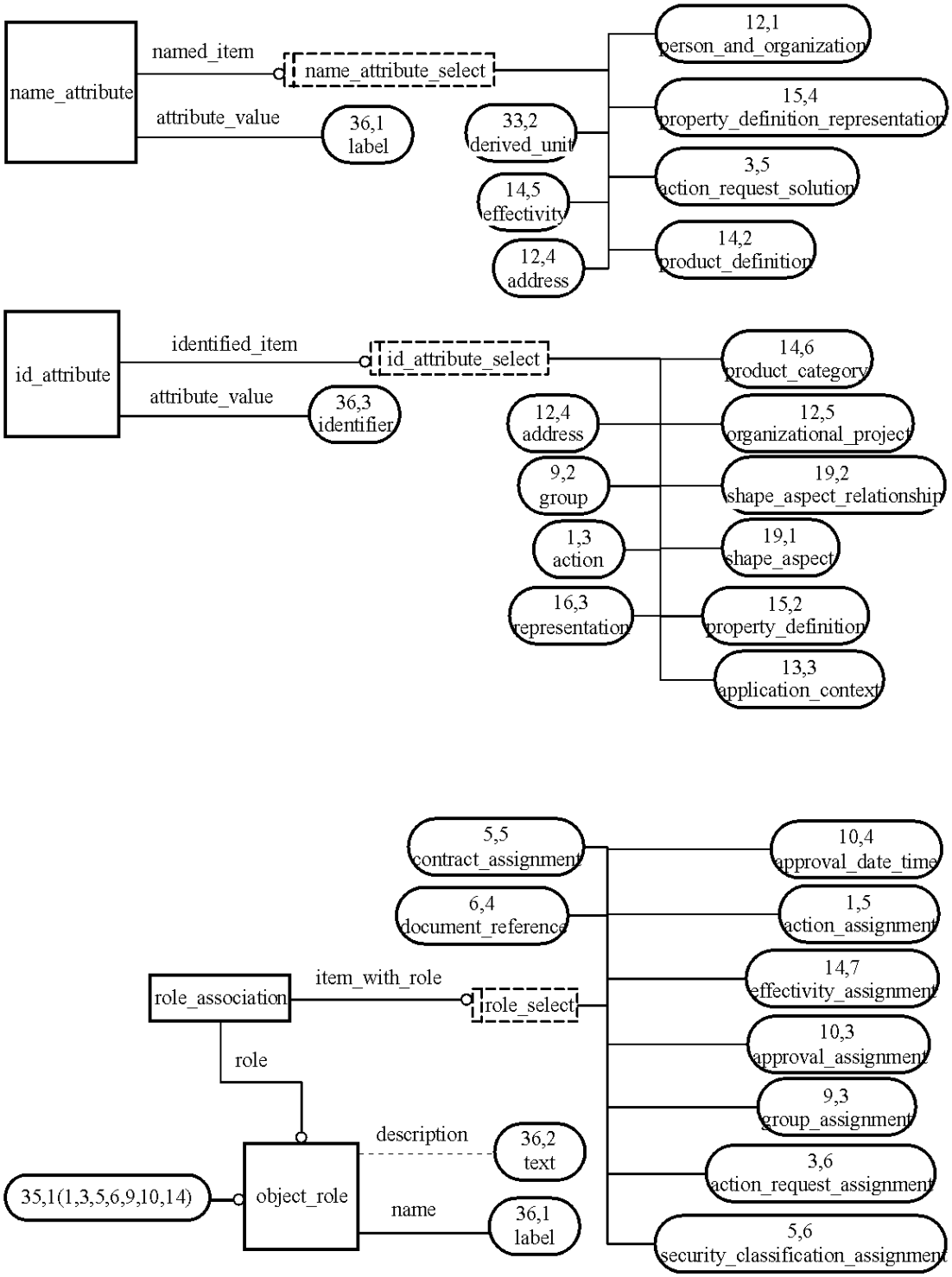


**Figure H.35 — attributes - AIM EXPRESS-G diagram 35 of 36**

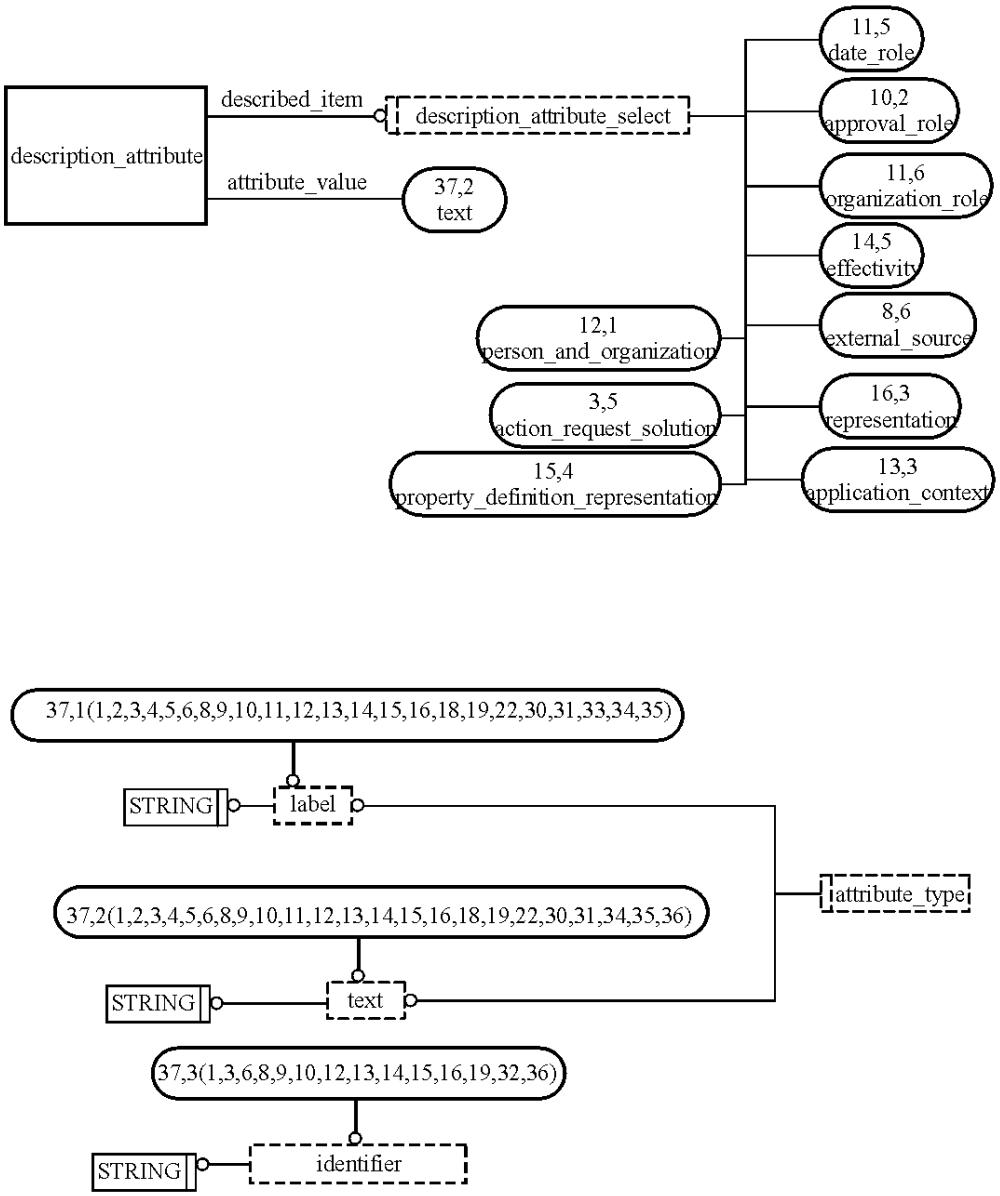


**Figure H.36 — description attribute - AIM EXPRESS-G diagram 36 of 36**

 

# Annex J
(informative)

# Computer interpretable listings

It also provides a listing of each EXPRESS schema specified in this part of ISO 10303 without comments or other explanatory text. These listings are available in computer-interpretable form and can be found at the following URLs:

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/SC4 Secretariat directly at: 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 K
(informative)

# Technical discussions

## L.1 Process_plan_version

A Process_plan_version defines the collection of manufacturing tasks or processes required to create a physical instance of a product as defined by a specified version of the product definition. The majority of manufacturing tasks utilized in the execution of the Process_plan_version typically employ Numerical Control (NC) or conventional (non-NC) mechanical material removal technologies. The Process_plan_-version identifies the task description and execution order, conditions of task execution, the manufacturing resources required by each task, and descriptive instructions for each task. At the most detailed level, the definition of the process(es) provided by the Process_plan_version is sufficient to define and control expected shop floor activities. It is not intended to satisfy a requirement to record or archive actual shop floor activities performed.

The information specified by a Process_plan_version can be at any level of detail required by the company. The information detail required for shop floor execution of the Process_plan_version is that level required to enable any assigned operator, with access to the designated manufacturing resources and without prior knowledge of the product/process definition, to unambiguously complete the designated task or manufacturing execution path.

The assigned operator is responsible for, and trained in, the safe and effective utilization of a designated manufacturing resource. The manufacturing resource typically associated with this definition in the context of this AP is an electrically powered, processor controlled tool. The person is knowledgeable of company specific operating practices.

The Process_plan_version will be composed of zero, one, or more activities required to define the manufacturing tasks required. A Process_plan_version with zero activities defined will be the typical situation at the beginning of the manufacturing planning function where a requirement has been identified, but the specific details have not been instantiated. A Process_plan_version may receive status or approval information as defined by business unit operating or configuration management procedures for the manufacturing planning function. The Status_authority for a specific Process_plan_version will accommodate the requirement to convey information such as “Responsible Planner: John Doe, Aug 12, 2003" or “Released for Production: Jane Doe, Aug 14, 2003". There can be multiple Status_authority items applicable to a given Process_plan_version at any one time. This is reflected by the current practice of multiple signature blocks on a process plan.

 

## Bibliography

[1] *Federal Information Processing Standards Publication 183, Integration Definition for Functional Modeling (IDEF0),* FIPS PUB 183, National Institute of Standards and Technology, December 1993.

[2] ISO 14649-10:2003 , *Industrial automation systems and integrations — Physical device control — Data model for computerized numerical controllers — Part 10: General process data.*

[3] ISO/CD 10303-238: — [1)]: *Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Application interpreted model for computerized numerical controllers.*

1) To be published

# Index



 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

 

**ICS 25.040.40**

Price based on 463 pages