INTERNATIONAL STANDARD

ISO 10303-224

Third edition
2006-08-15

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

Part 224: Application protocol: Mechanical product definition for process planning using machining feature

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

Partie 224: Protocole d'application: Définition de produits mécaniques pour la planification de procédés utilisant des caractéristiques d'usinage

Reference number
ISO 10303-224:2006(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

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 10303-224 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 4 *Industrial data*.

This third edition of ISO 10303-224 cancels and replaces the second edition (ISO 10303-224:2001), of which it constitutes a technical revision.

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

Each part of ISO 10303 is a member of one of the following series: description methods, implementation methods, conformance testing methodology and framework, integrated generic resources, integrated application resources, application protocols, abstract test suites, application interpreted constructs, and application modules. This part 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 production information and exchange of product data. The objective is to provide a neutral mechanism capable of describing product data throughout the life cycle of a product, independent from any particular system. The nature of this description makes it suitable not only for neutral file exchange, but also as a basis for implementing and sharing product databases and archiving.

ISO 10303 is organized as a series of parts, each published separately. The parts of ISO 10303 fall into one of the following series: description methods, integrated resources, application protocols, application interpreted constructs, abstract test suites, implementation methods, and conformance testing. The series are described in ISO 10303-1. This part of ISO 10303 is a member of the application protocol series.

ISO 10303-224 specifies an application protocol (AP) for the representation of information needed to produce a mechanical part definition for process planning of a single piece or an assembly of piece parts for machining operations, and specifies the integrated resources necessary to satisfy these requirements.

This application protocol defines the context, scope, and information requirements for the representation of information needed to produce a mechanical part definition. These requirements specify the part identification, tracking, shape, representation of the shape and material data necessary for the definition of a part for process planning. The process planning function in an organization can be assisted a great deal by identifying machining-oriented part shape features, so that the process planner can more readily identify machining tools and processes to manufacture a part.

This application protocol identifies specific characteristics of part shape used in manufacturing. These characteristics are used to define manufacturing features. These shapes can be represented either by machining features defined in this application protocol, or by a boundary representation solid model, shared by other application protocols and used as application interpreted constructs in this part. The purpose of manufacturing features is to facilitate the identification of manufacturing shapes that are human and computer interpretable. Manufacturing features allow information about the shape to be used for decisions in computerized process planning systems.

Information about the part material is supplied so the process planning activity can determine equipment and material requirements. Also supplied is administrative information necessary for tracking customer information, supplier information about the part, and internal control information for the manufacturing operation to support process planning. Tracking of certain administrative information is a component of the iterative process for creating a process plan.

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. This planning model was created from the in-scope data from the activities of the application activity model (AAM) and grouped into logical units of functionality. This planning model is used as a guide in developing the application reference model (ARM).

This edition of this part of ISO 10303 (ISO 10303-224:2006) incorporates modifications that are upwardly compatible with the previous edition, and modifications that are not upwardly compatible with the previous edition. Modifications to EXPRESS specifications are upwardly compatible if:

— the modifications do not result in changes to instances that are encoded according to ISO 10303-21; such instances conform to both the unmodified and modified EXPRESS specifications;

— the modifications do not result in changes to software that conforms to ISO 10303-22 with respect to access to the data content of data structures;

— the modifications do not invalidate mappings to the previous edition of this part of ISO 10303 that are specified in the mapping table of an ISO 10303 application protocol.

Technical modifications are summarized in Annex K.

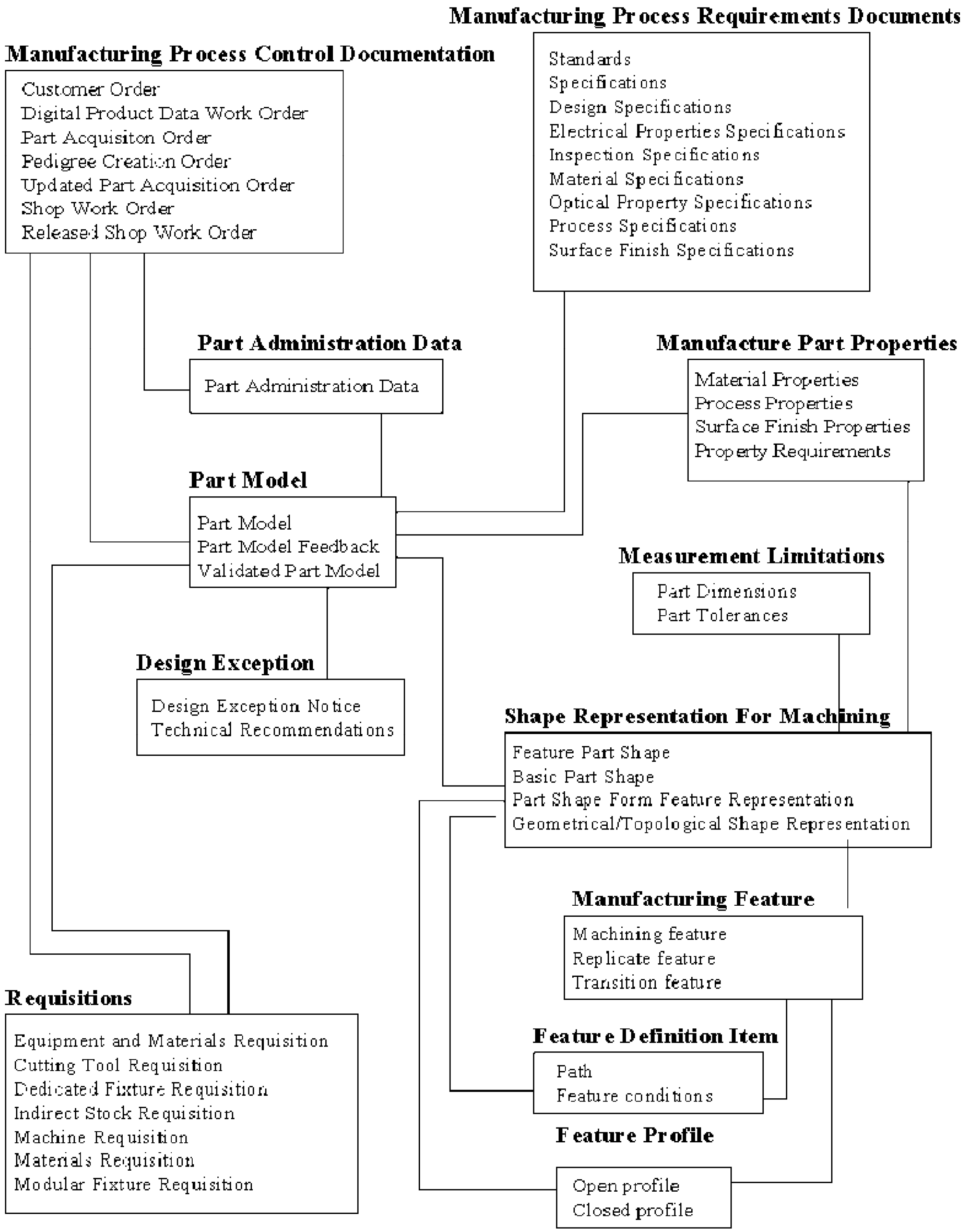


Figure 1 — Data planning model

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

Part 224:
Application protocol: Mechanical product definition for process planning using machining features

1 Scope

This part of ISO 10303 specifies the information requirements for the representation and exchange of information needed to define the product data which is necessary in the manufacturing of a single piece or assembly of mechanical parts. It also specifies the integrated resources necessary to satisfy these requirements. The product data is based on existing part designs whose shapes are represented by machining features. This part of ISO 10303 supports digital representation for computer integrated manufacturing.

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

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

— the manufacture of a single piece mechanical part, and assemblies of single piece parts for manufacturing purposes;

— parts that are to be manufactured by either milling or turning processes;

— machining features for defining shapes necessary for manufacturing;

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

— explicit representation of the 3D shape of machining features through bounded geometry representations;

— implicit representation of machining features through selection of standard parameters;

— machining feature definition elements necessary for creating machining form features;

— customer order administrative data to track receipt of an order for a part to the shop floor, but not including tracking of the order on the shop floor;

— approval data to authorize the manufacture of a part;

— requisition administrative data to identify requirements and track the status of materials and equipment needed to manufacture a part;

— work order data to track and identify the status of a part;

— tracking the state of raw stock for documenting the manufacturing history of a part;

— tracking a design exception notice of a part.

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

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

— results from process planning functions;

— exchange of data within process planning systems;

— feature order or sequence;

— representation of assemblies for design or bill of materials;

— representation of composite material parts;

— representation of sheet metal parts;

— representation of part pedigree;

— design features of a part;

— schedule for completing a work order through the manufacturing process;

— configuration control for a part.

2 Normative references

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

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

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

ISO 1101:2004, *Geometrical Product Specification (GPS) — Geometrical tolerancing — Tolerances of form, orientation, location and run-out*

ISO 1122-1:1998, *Vocabulary of gear terms — Part 1: Definitions related to geometry*

ISO 1328-1, *Cylindrical gears — ISO system of accuracy — Part 1: Definitions and allowable values of deviations relevant to corresponding flanks of gear teeth*

ISO 1340, *Cylindrical gears — Information to be given to the manufacturer by the purchaser in order to obtain the gear required*

ISO 2203, *Technical drawings — Conventional representation of gears*

ISO 5459:1981, *Technical drawings — Geometrical tolerancing — Datums and datum-systems for geometrical tolerances*

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

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

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

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

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

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

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

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

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

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

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

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

3 Terms, definitions, and abbreviations

3.1 Terms defined in ISO 1101

For the purpose of this part of ISO 10303, the following terms defined in ISO 1101 apply.

— dimension;

— tolerance.

3.2 Terms defined in ISO 5459

For the purpose of this part of ISO 10303, the following terms defined in ISO 5459 apply.

— datum.

3.3 Terms defined in ISO 10303-1

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

— application;

— application activity model (AAM);

— application interpreted model (AIM);

— application object;

— application protocol (AP);

— application reference model (ARM);

— implementation method;

— integrated resource;

— model;

— product;

— product data;

— protocol implementation conformance statement (PICS);

— unit of functionality (UoF).

3.4 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.

3.5 Terms defined in ISO 10303-42

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-42 apply.

— boundary representation (B-rep) solid model;

— manifold solid boundary representation.

3.6 Terms defined in ISO 10303-240

For the purpose of this part of ISO 10303, the following terms defined in ISO 10303-240 apply.

— process plan;

— workstation.

3.7 Other definitions

For the purpose of this part of ISO 10303, the following definitions apply.

3.7.1
basic angle
centre lines of features in a pattern or surfaces shown on a drawing, located or defined by the use of basic dimensions, where no angle is specified

3.7.2
cutting tool
instrument used to remove material from a part

3.7.3
dedicated fixture
fixture designed for a particular part

3.7.4
fixture
device to hold and locate a part during manufacturing and inspection

3.7.5
indirect material
consumable material used during the manufacture of a part

EXAMPLES Oil, cutting fluid, pallets, brooms, shop towels

3.7.6
machine
structure consisting of a framework with various moving parts, used for material removal operations to a part

3.7.7
modular fixture
fixture component that is interchangeable and can be assembled into different configurations as needed

3.7.8
raw stock
any material that can have a manufacturing process applied, including castings and forgings

3.8 Abbreviations

For the purpose 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
B-rep	boundary representation
ICOM	input, control, output, mechanism
Ngon	N number of sides polygon
PICS	protocol implementation conformance statement
UoF	unit of functionality

4 Information requirements

This clause specifies the information required for the definition of product data for mechanical product definition, in the context of process planning using machining features.

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 in the scope of the application protocol in Annex F.

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

4.1 Units of functionality

This subclause specifies the units of functionality for the mechanical product definition for process planning using machining features application protocol. This part of ISO 10303 specifies the following units of functionality:

— design_exception;

— feature_definition_item;

— feature_profile;

— library_reference;

— manufacturing_feature;

— manufacturing_part_properties;

— manufacturing_process_control_documentation;

— manufacturing_process_requirement_documents;

— measurement_limitations;

— part_administration_data;

— part_model;

— requisitions;

— shape_representation_for_machining.

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 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.

4.1.2 feature_definition_item

The feature_definition_item UoF contains the information necessary to create a machining feature. Additionally this UoF identifies the relationship between machining features and aspects of shape.

The following application objects are used by the feature_definition_item UoF:

— Angle_taper;

— Blind_bottom_condition;

— Boss_top_condition;

— Chamfer_angle;

— Circular_path;

— Complete_circular_path;

— Conical_hole_bottom;

— Diameter_taper;

— Directed_taper;

— First_offset;

— Flat_hole_bottom;

— Flat_slot_end_type;

— Flat_with_radius_hole_bottom;

— Flat_with_taper_hole_bottom;

— General_path;

— General_pocket_bottom_condition;

— General_profile_floor;

— General_rib_top_floor;

— General_top_condition;

— Linear_path;

— Loop_slot_end_type;

— Open_slot_end_type;

— Partial_area_definition;

— Partial_circular_path;

— Path;

— Planar_pocket_bottom_condition;

— Planar_profile_floor;

— Planar_rib_top_floor;

— Planar_top_condition;

— Pocket_bottom_condition;

— Profile_floor;

— Radiused_slot_end_type;

— Rib_top_floor;

— Second_chamfer_offset;

— Second_offset;

— Slot_end_type;

— Spherical_hole_bottom;

— Thread_runout;

— Through_bottom_condition;

— Through_pocket_bottom_condition;

— Through_profile_floor;

— Woodruff_slot_end_type.

4.1.3 feature_profile

The feature_profile UoF contains the information necessary to identify 2D shapes. By sweeping a feature_profile along a path, 3D features are created.

The following application objects are used by the feature_profile UoF:

— Circular_closed_profile;

— Closed_profile;

— General_closed_profile;

— General_open_profile;

— Linear_profile;

— Ngon_profile;

— Open_profile;

— Partial_circular_profile;

— Profile;

— Rectangular_closed_profile;

— Rounded_U_profile;

— Square_U_profile;

— Tee_profile;

— Vee_profile.

4.1.4 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 can 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.

4.1.5 manufacturing_feature

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

The following application objects are used by the manufacturing_feature UoF:

— Bevel_gear;

— Boss;

— Catalogue_gear;

— Catalogue_knurl;

— Catalogue_marking;

— Catalogue_thread;

— Chamfer;

— Circular_boss;

— Circular_closed_shape_profile;

— Circular_cutout;

— Circular_offset_pattern;

— Circular_omit_pattern;

— Circular_pattern;

— Closed_slot;

— Compound_feature;

— Compound_feature_element;

— Compound_feature_relationship;

— Constant_radius_edge_round;

— Constant_radius_fillet;

— Counterbore_hole;

— Countersunk_hole;

— Cutout;

— Defined_gear;

— Defined_marking;

— Defined_thread;

— Diagonal_knurl;

— Diamond_knurl;

— Edge_round;

— Fillet;

— Gear;

— General_boss;

— General_cutout;

— General_outside_profile;

— General_pattern;

— General_pocket;

— General_removal_volume;

— General_revolution;

— General_shape_profile;

— Groove;

— Helical_gear;

— Helical_bevel_gear;

— Hole;

— Knurl;

— Machining_feature;

— Manufacturing_feature;

— Manufacturing_feature_group;

— Marking;

— Multi_axis_feature;

— Open_slot;

— Outer_diameter;

— Outer_diameter_to_shoulder;

— Outer_round;

— Partial_circular_shape_profile;

— Planar_face;

— Pocket;

— Profile_feature;

— Protrusion;

— Recess;

— Rectangular_boss;

— Rectangular_closed_pocket;

— Rectangular_closed_shape_profile;

— Rectangular_offset_pattern;

— Rectangular_omit_pattern;

— Rectangular_open_pocket;

— Rectangular_open_shape_profile;

— Rectangular_pattern;

— Replicate_base;

— Replicate_feature;

— Revolved_feature;

— Revolved_flat;

— Revolved_round;

— Rib_top;

— Round_hole;

— Rounded_end;

— Shape_profile;

— Slot;

— Spherical_cap;

— Spur_gear;

— Step;

— Straight_bevel_gear;

— Straight_knurl;

— Thread;

— Transition_feature;

— Turned_knurl.

4.1.6 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.

4.1.7 manufacturing_process_control_documentation

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

The following application objects are used by the manufacturing_process_control_documentation UoF:

— Customer_order;

— Digital_technical_data_package_work_order;

— Ordered_part;

— Pedigree_creation_order;

— Project_order;

— Resource_acquisition_order;

— Shop_work_order.

4.1.8 manufacturing_process_requirement_documents

The manufacturing_process_requirement_documents UoF provides the ability to specify documents that are directly related to product data as they support the definition of the product. These documents may be specific to an operation on the part being manufactured or a property of the part at a particular stage in the manufacturing process. The documents may, but need not be, company, customer, military, national, or international standards.

The following application objects are used by the manufacturing_process_requirement_documents UoF:

— Part_dimensioning_standard;

— Specification;

— Specification_usage_constraint.

4.1.9 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;

— Angularity_tolerance;

— Circular_runout_tolerance;

— Circularity_tolerance;

— Compound_datum;

— Concentricity_tolerance;

— Curved_dimension_tolerance;

— Cylindricity_tolerance;

— Datum;

— Datum_feature;

— Datum_target;

— Datum_target_set;

— Diameter_dimension_tolerance;

— Dimensional_tolerance;

— Distance_along_curve_tolerance;

— Externally_defined_size_dimension;

— Flatness_tolerance;

— Geometric_tolerance;

— Geometric_tolerance_precedence_relationship;

— Height_dimension;

— Length_dimension;

— Limits_and_fits;

— Linear_profile_tolerance;

— Location_dimension_tolerance;

— Location_tolerance;

— Material_condition_modifier;

— Parallelism_tolerance;

— Perpendicularity_tolerance;

— Placed_target;

— Plus_minus_value;

— Position_tolerance;

— Projection;

— Radial_dimension_tolerance;

— Size_tolerance;

— Straightness_tolerance;

— Surface_profile_tolerance;

— Symmetry_tolerance;

— Target_area;

— Target_circle;

— Target_line;

— Target_point;

— Target_rectangle;

— Thickness_tolerance;

— Tolerance_limit;

— Tolerance_range;

— Tolerance_value;

— Tolerance_zone;

— Tolerance_zone_definition;

— Total_runout_tolerance;

— Width_dimension.

4.1.10 part_administration_data

The part_administration_data UoF contains the information used in the management of product data.

The following application objects are used by the part_administration_data UoF:

— Approval;

— Organization;

— Person;

— Person_in_organization.

4.1.11 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 object is used by the part_model UoF:

— Manufactured_assembly;

— Manufactured_assembly_relationship;

— Mating_definiton;

— Mating_definition_relationship;

— Mating_relationship;

— Part;

— Single_piece_part.

4.1.12 requisitions

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

The following application objects are used by the requisitions UoF:

— Cutting_tool_requisition;

— Dedicated_fixture_requisition;

— Indirect_stock_requisition;

— Machine_requisition;

— Material_requisition;

— Modular_fixture_requisition;

— Requisition.

4.1.13 shape_representation_for_machining

The shape_representation_for_machining UoF contains the physical definition of initial and final form of the part. This definition is given via a parametric method for features, and additional geometric and topological definitions.

The following application objects are used by the shape_representation_for_machining UoF:

— Base_shape;

— Block_base_shape;

— B-rep_model;

— B-rep_model_element;

— B-rep_shape_aspect_representation;

— B-rep_shape_representation;

— Cylindrical_base_shape;

— Direction_element;

— Explicit_base_shape_representation;

— Face_shape_element;

— Face_shape_element_relationship;

— Implicit_base_shape_representation;

— Location_element;

— Ngon_base_shape;

— Orientation;

— Part_placement;

— Path_element;

— Planar_element;

— Shape;

— Shape_aspect;

— Shape_element.

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 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:

— alternate_ranking;

— material_substitute.

4.2.1.1 alternate_ranking

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

4.2.1.2 material_substitute

The material_substitute specifies the material to be used as the substitute. See 4.3.2 for the application assertion.

4.2.2 Angle_taper

An Angle_taper is a constant change in shape of a feature for a part. The start of the Angle_taper is at the placement of a feature and is applied to the entire feature. The length of the taper is determined from the feature that is applying the Angle_taper.

NOTE Figure 2 illustrates features that have an Angle_taper applied.

The data associated with an Angle_taper are the following:

— angle.

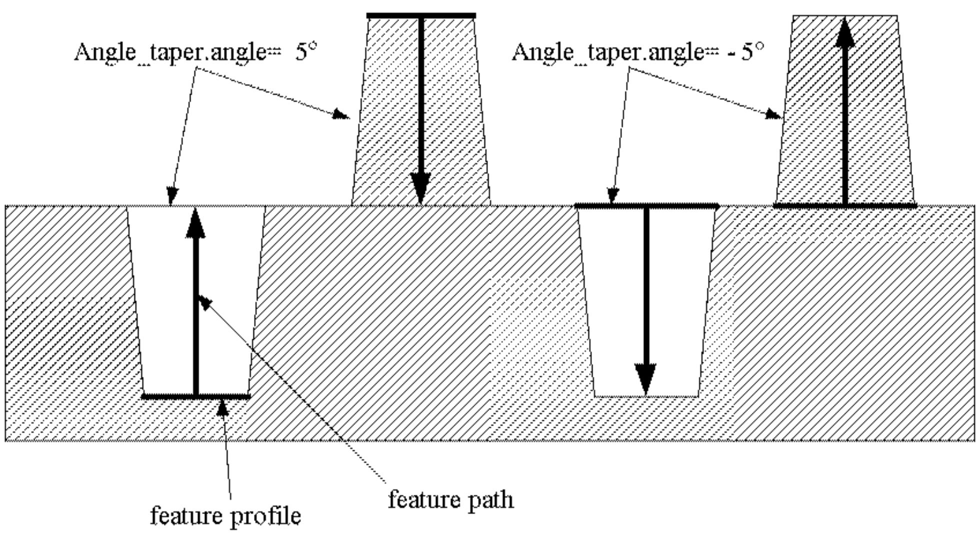


Figure 2 — Angle_taper

4.2.2.1 angle

The angle specifies the amount of slope from the start of an Angler_taper to the end of the Angle_taper. An angle between 0 and 90 degrees or between -90 and -180 degrees indicates that the profile for a feature grows larger along the feature path. An angle between 0 and -90 degrees or between 90 and 180 degrees indicates that the profile for a feature grows smaller along thefeature path. See 4.3.3 for the application assertion.

4.2.3 Angular_dimension_tolerance

An Angular_dimension_tolerance is a type of Location_tolerance (see 4.2.123) 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 Figure 3 illustrates the Angular_dimension_tolerance.

The data associated with an Angular_dimension_tolerance are the following:

— plane_and_direction.

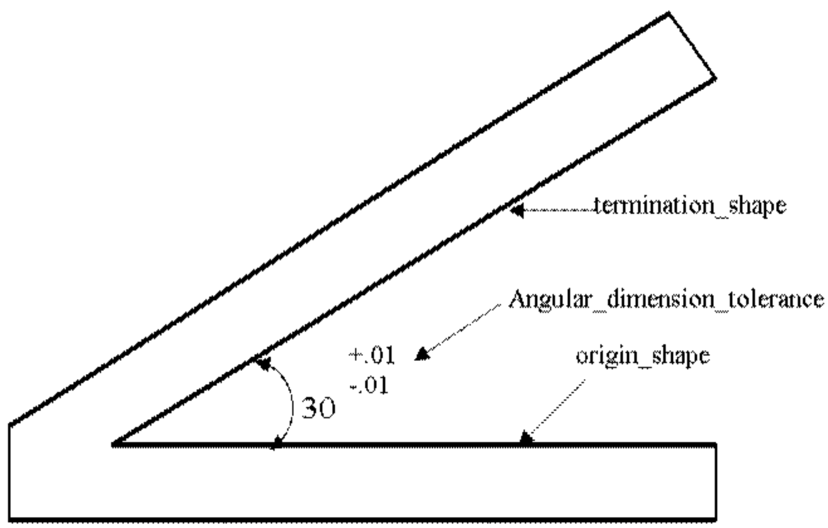


Figure 3 — Angular_dimension_tolerance

4.2.3.1 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 4.3.4 for the application assertion.

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

4.2.4 Angular_size_dimension_tolerance

An Angular_size_dimension_tolerance is a type of Size_tolerance (see 4.2.223) that specifies the allowable variation on the size or gap formed by two angular elements of the shape of a part.

NOTE Figure 4 illustrates the Angular_size_dimension_tolerance.

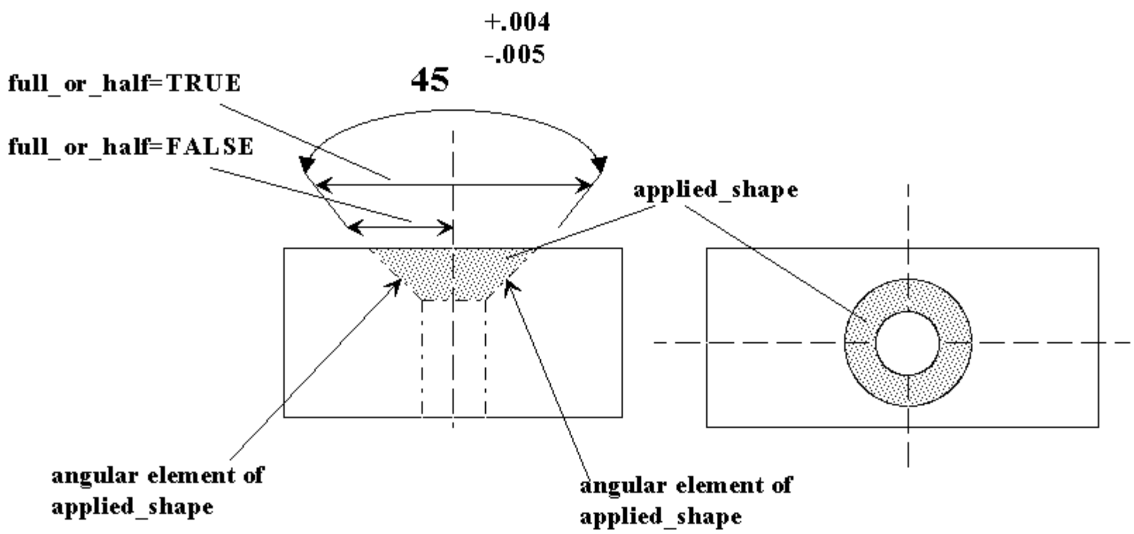


Figure 4 — Angular_size_dimension_tolerance

The data associated with an Angular_size_dimension_tolerance are the following:

— full_or_half;

— major_angle.

4.2.4.1 full_or_half

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

4.2.4.2 major_angle

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

4.2.5 Angularity_tolerance

An Angularity_tolerance is a type of Geometric_tolerance (see 4.2.104) that is the allowable variation of a surface or axis at a specified angle (other than 90 degrees) from a datum plane or axis. An Angularity_-tolerance constrains a shape that is one of the following:

— An allowable variation, defined by a tolerance zone, between a surface and a datum plane specified by the basic angle;

— An allowable variation, defined by a diametric tolerance zone, between an axis and a datum plane specified by the basic angle.

NOTE 1 Figure 5 illustrates the Angularity_tolerance for a plane surface.

NOTE 2 Figure 6 illustrates Angularity_tolerance for an axis.

NOTE 3 The Angularity_tolerance definition is derived from ISO 1101:2004, 18.11.

The data associated with an Angularity_tolerance are the following:

— geometric_reference;

— segment_size.

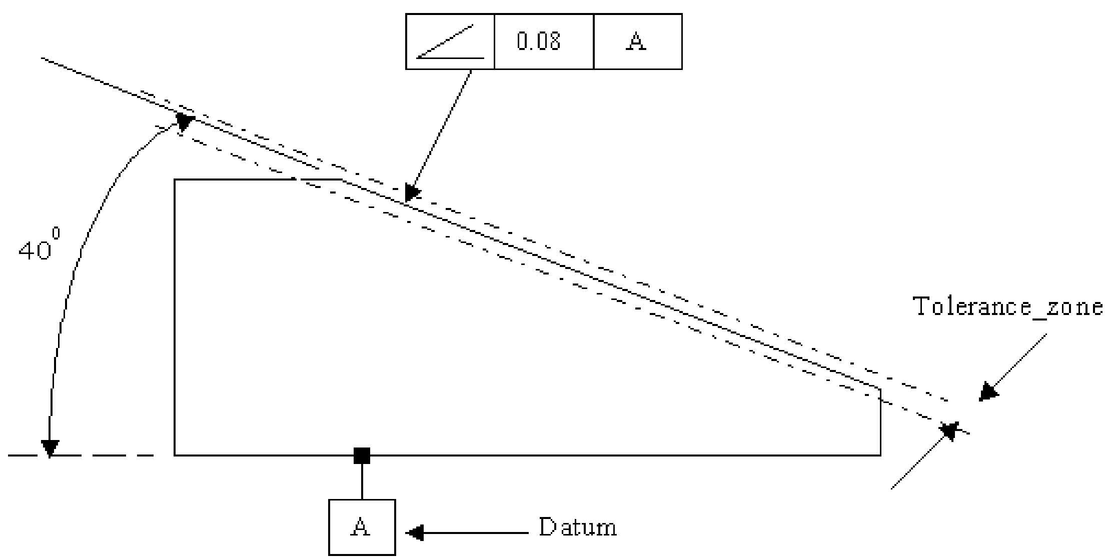


Figure 5 — Angularity_tolerance for a plane surface

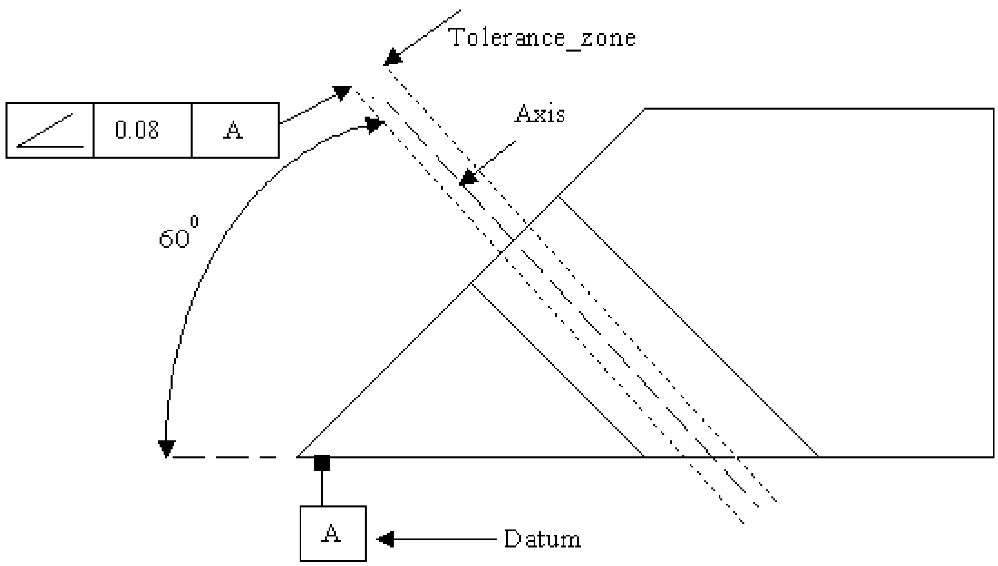


Figure 6 — Angularity_tolerance for an axis

4.2.5.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.5 for the application assertion.

4.2.5.2 segment_size

The segment_size specifies the length of a surface to apply a tolerance if the Angularity_tolerance is not applied to the total length. The segment_size need not be specified for a particular Angularity_tolerance.

4.2.6 Approval

An Approval is the indication within an organization of concurrence or nonconcurrence with a piece of product data. The data associated with an Approval are the following:

— approval_authority;

— approval_date;

— status.

4.2.6.1 approval_authority

The approval_authority specifies the responsible person and company for making the approval. There may be more than one approval_authority for an Approval. See 4.3.6 for the application assertion.

4.2.6.2 approval_date

The approval_date specifies a calendar date on which the approval status was set.

4.2.6.3 status

The status specifies the state of consent applied to the approval to manufacture a part.
The values of the status may be one of the following:

— approved;

— disapproved;

— not_yet_approved;

— withdrawn.

NOTE See 4.2.6.3.1 - 4.2.6.3.4 for the definition of each allowable value for status.

4.2.6.3.1 approved:
a concurrence has been given on a specific date.

4.2.6.3.2 disapproved:
a nonconcurrence has been given on a specific date.

4.2.6.3.3 not_yet_approved:
a concurrence has not yet been given.

4.2.6.3.4 widthdrawn:
a concurrence has been removed on a specific date.

4.2.7 Base_shape

A Base_shape is the bounded geometry for the part before machining features are applied. The Base_shape defines the form and minimum dimensions required to determine the raw stock (material) required to be issued for the manufacture of a part. Each Base_shape is either an Explicit_base_-shape_representation (see 4.2.76) or an Implicit_base_shape_representation (see 4.2.112).

4.2.8 Bevel_gear

A Bevel_gear is a type of Defined_gear (see 4.2.59) that is used in meshing pairs where the axes of the pairs intersect. Each Bevel_gear is either a Helical_bevel_gear (see 4.2.110), or a Straight_bevel_gear (see 4.2.233). The Bevel_gear definition is based upon the shaft angle which is defined as the smallest angle through which one of the axes must be rotated to be parallel. This angle is a critical part of bevel gear design. For a gear design, the shaft angle is required for defining the pitch angle from which the tip_angle and root_angle are defined.

NOTE Figure 7 illustrates the Bevel_gear, and Figure 8 illustrates the shaft angle, and pitch angle.

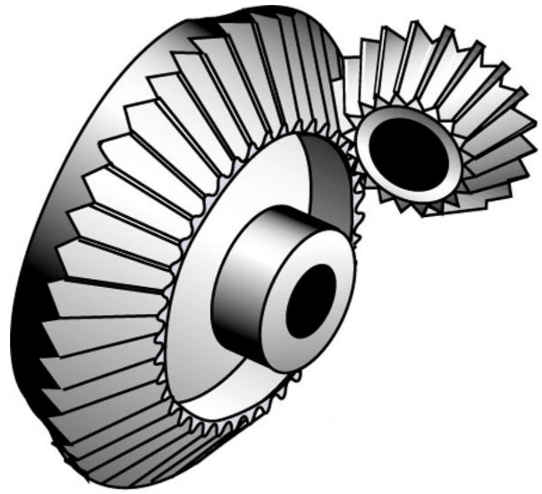


Figure 7 — Bevel_gear

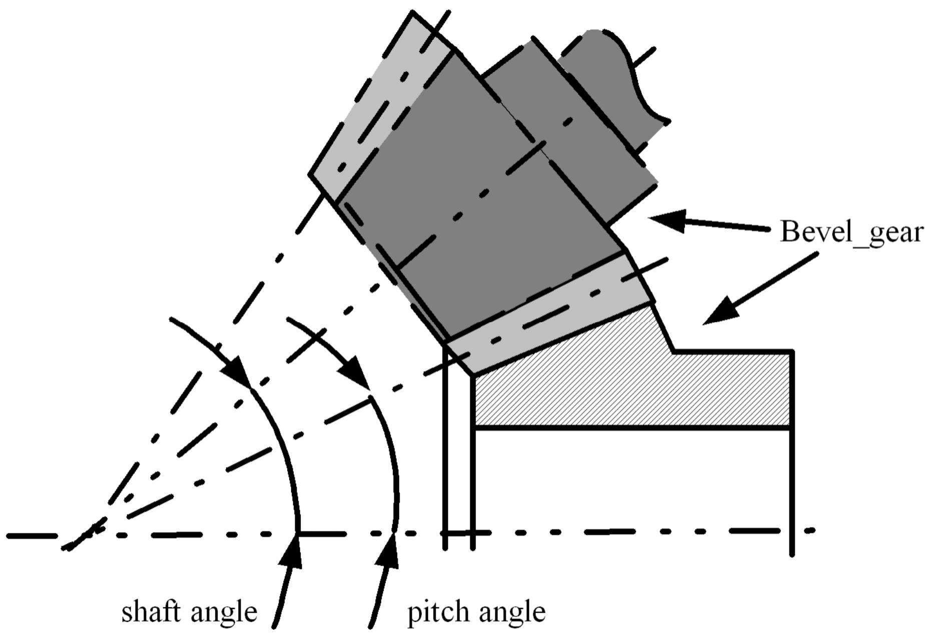


Figure 8 — Bevel_gear with shaft angle, and pitch angle

The data associated with a Bevel_gear are the following:

— root_angle;

— tip_angle.

4.2.8.1 root_angle

The root_angle specifies the angle between the axis and the root of the teeth. See 4.3.7 for the application assertion.

4.2.8.2 tip_angle

The tip_ the angle specifies between the axis and the tip of the teeth. See 4.3.7 for the application assertion.

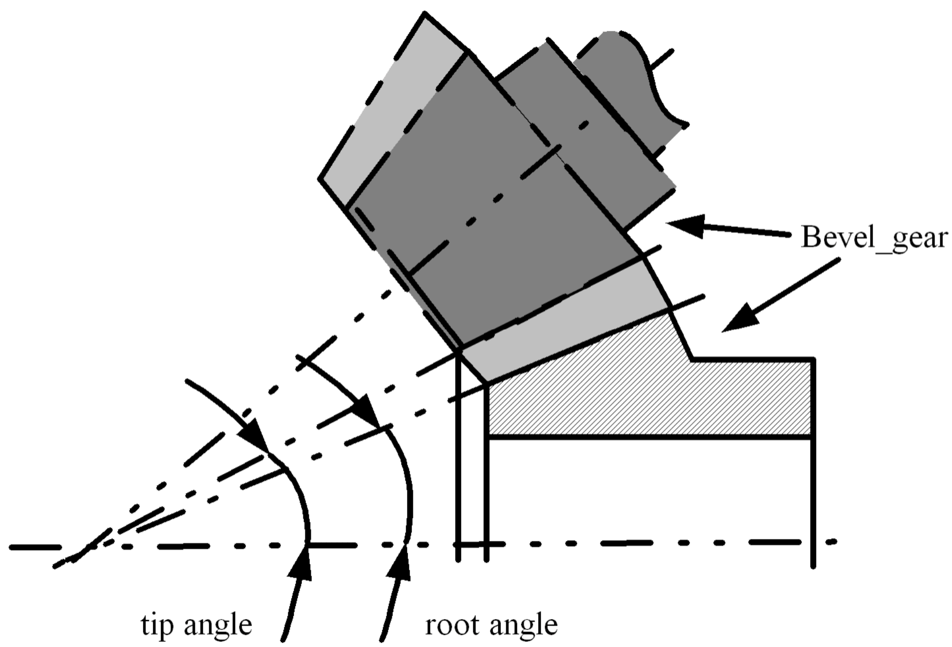


Figure 9 — Bevel_gear with root_angle and tip_angle

4.2.9 Blind_bottom_condition

A Blind_bottom_condition is a selection type of Hole_bottom_condition_select that has material in the bottom of a hole and does not go through the entire part. Each Blind_bottom_condition is either a Conical_hole_bottom (see 4.2.43), a Flat_hole_bottom (see 4.2.83), a Flat_with_radius_hole_bottom (see 4.2.85), a Flat_with_taper_hole_bottom (see 4.2.86), or a Spherical_hole_bottom (see 4.2.229).

The data associated with a Blind_bottom_condition are the following:

— start_or_end.

4.2.9.1 start_or_end

The start_or_end specifies a boolean value of TRUE if the Blind_bottom_condition is positioned at the end of a Round_hole, and a value of FALSE if it is at the start of the Round_hole.

4.2.10 Block_base_shape

A Block_base_shape is a type of Implicit_base_shape_representation (see 4.2.112) that describes the initial shape of the material as a rectangular cross section of some determined length.

NOTE Figure 10 illustrates a Block_base_shape and the associated attributes.

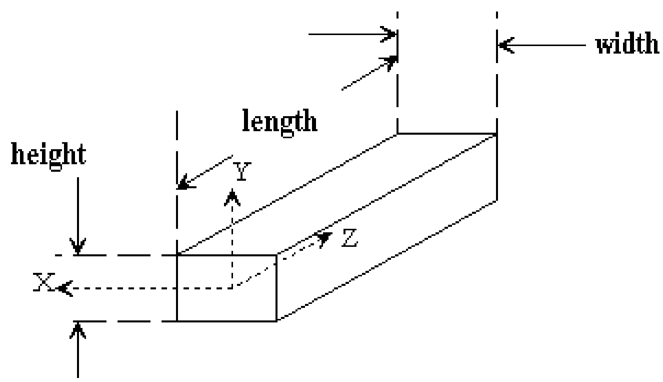


Figure 10 — Block_base_shape

The data associated with a Block_base_shape are the following:

— height;

— width.

4.2.10.1 height

The height specifies the size of the side of a Block_base_shape along the Y-axis. See 4.3.8 for the application assertion.

4.2.10.2 width

The width specifies the size of the side of a Block_base_shape along the X-axis. See 4.3.8 for the application assertion.

4.2.11 Boss

A Boss is a type of Multi_axis_feature(see 4.2.140) that is a closed shape that protrudes from the surface of the part. The intersection of the Boss and the base surface may have a radius shaped blend between them. Each Boss is either a Circular_boss (see 4.2.24), a General_boss (see 4.2.89), or a Rectangular_boss (see 4.2.194). The Boss may be positioned on the face of a part with the Z-axis in the direction away from the part, or at the top of the Boss with the Z-axis in the direction toward the part face.

The data associated with a Boss are the following:

— boss_height;

— fillet_radius;

— top_condition.

4.2.11.1 boss_height

The boss_height specifies the maximum height of a Boss measured from the highest point on the Boss to the farthest point of intersection between the Boss and the part surface. The placement and orientation of the Linear_path shall be the same as the Boss feature. See 4.3.10 for the application assertion.

4.2.11.2 fillet_radius

The fillet_radius specifies a radius shape blend between a Boss and the surrounding surface at the base of the Boss. See 4.3.10 for the application assertion.

4.2.11.3 top_condition

The top_condition specifies the shape of the top of a Boss feature. See 4.3.9 for the application assertion.

4.2.12 Boss_top_condition

A Boss_top_condition is the end shape of a Boss that is the farthest distance away from the intersection of the face of the part and the Boss. A Boss top may be either flat or of any other shape. Each Boss_-top_condition is either a General_top_condition (see 4.2.103) or a Planar_top_condition (see 4.2.175). The data associated with a Boss_top are the following:

— start_or_end;

— top_radius.

4.2.12.1 start_or_end

The start_or_end specifies a boolean value of TRUE if the Boss_top_condition is located at the start of the Boss, FALSE if it is located at the end of the Boss.

4.2.12.2 top_radius

The top_radius specifies a radius shape blend between a Boss_top_condition and the surrounding Boss surface at the top of the Boss. See 4.3.12 for the application assertion.

NOTE If the Boss_top_condition ia a General_top_condition that defines a spherical shape, the top_radius does not necessarily apply.

4.2.13 B-rep_model

A B-rep_model is a solid model containing complete representation of shape using manifold solid boundary representation.

4.2.14 B-rep_model_element

A B-rep_model_element is a portion of a boundary representation. The data associated with a B-rep_-model_element are the following:

— element.

4.2.14.1 element

The element specifies the portion of the B-rep_model that defines the B-rep_model_element. See 4.3.13 for the application assertion.

4.2.15 B-rep_shape_aspect_representation

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

— shape_definition.

4.2.15.1 shape_definition

The shape_definition specifies the shape that is the boundary representation. See 4.3.14 for the application assertion.

4.2.16 B-rep_shape_representation

A B-rep_shape_representation is a grouping of boundary representations to define a shape. The data associated with a B-rep_shape_representation are the following:

— shape_definition.

4.2.16.1 shape_definition

The shape_definition specifies the shape that is the boundary representation. See 4.3.15 for the application assertion.

4.2.17 BSU

A BSU identifies a piece of information, that can be a supplier, a class or a property, by specifying a code and a version. Each BSU is either a Supplier_BSU (see 4.2.237), a Class_BSU (see 4.2.34), or a Property_BSU (see 4.2.187).

The data associated with a BSU are the following:

— code.

4.2.17.1 code

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

4.2.18 Catalogue_gear

A Catalogue_gear is a is a type of Gear (see 4.2.88) that is a reference to a document containing the information to create a gear on a part.

NOTE Figure 11 illustrates a Catalogue_gear that is a 24 toothed gear with further definition found in a specification.

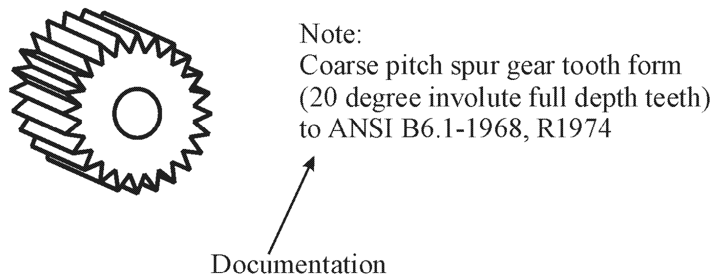


Figure 11 — Catalogue_gear

The data associated with a Catalogue_gear are the following:

— documentation.

4.2.18.1 documentation

The documentation specifies the document that defines information pertaining to a Gear feature. See 4.3.16 for the application assertion.

4.2.19 Catalogue_knurl

A Catalogue_knurl is a type of Knurl (see 4.2.114) that is a reference to a document containing the information to create a knurl on a part.

NOTE Figure 12 illustrates a Catalogue_knurl that is a diagonal knurl with further definition found in a specification called ANSI B94.6.

The data associated with a Catalogue_knurl are the following:

— documentation.

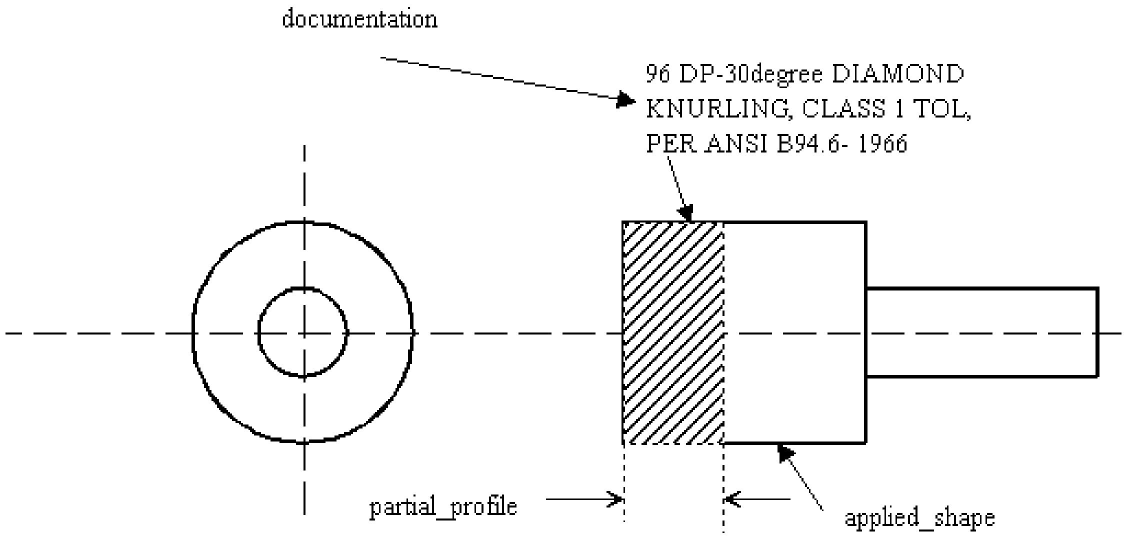


Figure 12 — Catalogue_knurl

4.2.19.1 documentation

The documentation specifies the document that defines information pertaining to a Knurl feature. See 4.3.17 for the application assertion.

4.2.20 Catalogue_marking

A Catalogue_marking is a type of Marking (see 4.2.131) that is a reference to a document containing the information for marking on a surface of a part.

NOTE Figure 13 illustrates a Catalogue_marking with a text value of '1928-12345ED34' a documentation value of 'ANSI/SAE-AS478-2 ', and associated data for the applied_shape.

The data associated with a Catalogue_marking are the following:

— documentation.

4.2.20.1 documentation

The documentation specifies the document or specification that defines information pertaining to a Marking feature. See 4.3.18 for the application assertion.

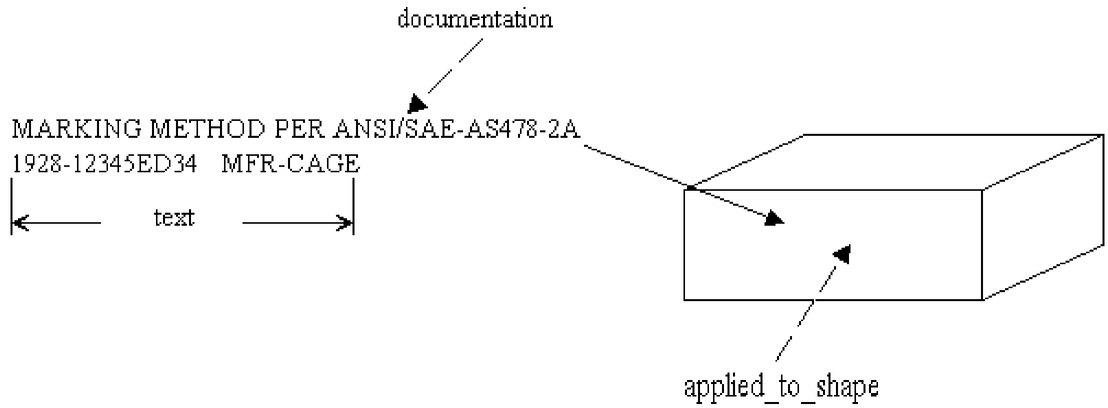


Figure 13 — Catalogue_marking

4.2.21 Catalogue_thread

A Catalogue_thread is a type of Thread (see 4.2.246) that is a reference to a document containing the information to create threads on a part.

NOTE Figure 14 illustrates a Catalogue_thread that is an outer thread with values for the documentation associated data.

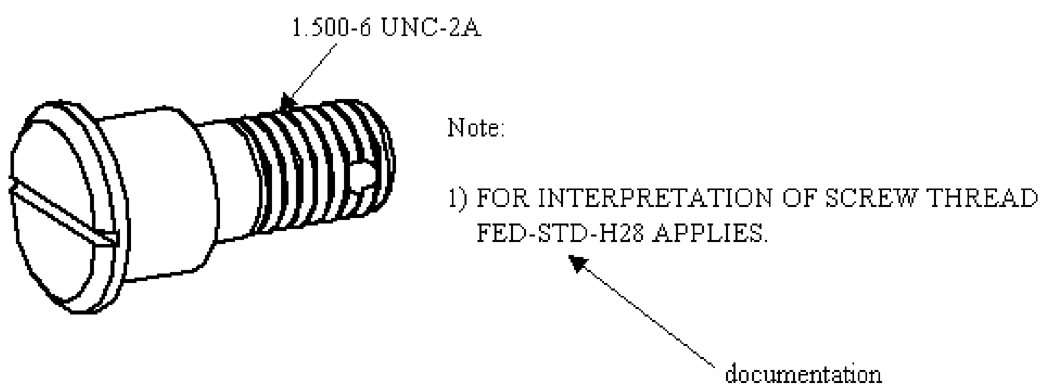


Figure 14 — Catalogue_thread

The data associated with a Catalogue_thread are the following:

— documentation;

— major_diameter.

4.2.21.1 documentation

The documentation specifies the document that defines information pertaining to a Thread feature. See 4.3.19 for the application assertion.

4.2.21.2 major_diameter

The major_diameter specifies the dimension of the largest diameter of the Thread and is applied to both an internal and an external thread. See 4.3.20 for the application assertion.

4.2.22 Chamfer

A Chamfer is a type of Transition_feature (see 4.2.258) that is a transition between corresponding edges of two joining non-coplanar surfaces, having a flat cross section. A Chamfer feature requires an offset length from one face, and either an angular amount from the same surface or an offset length from a second face.

NOTE Figure 15 illustrates a Chamfer applied to a rectangular block and the chamfer_face associated data. The offset_amout and angle_amout attribtes apply to the First_offset, Second_offset, Second_chamfer_offset and Chamfer_angle entities.

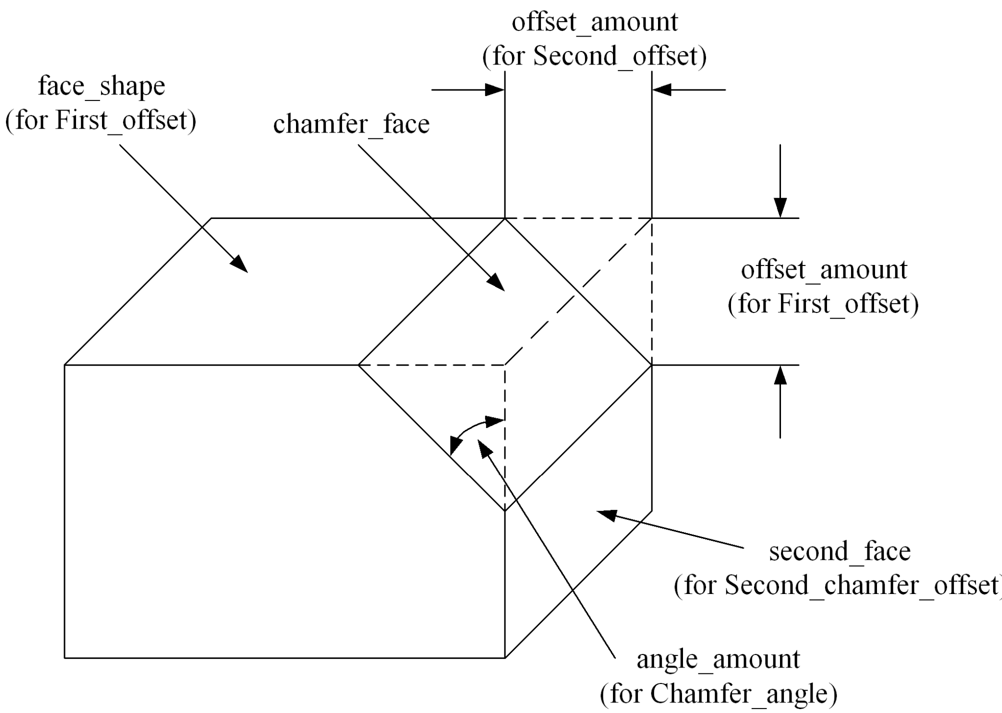


Figure 15 — Chamfer

The data associated with a Chamfer are the following:

— chamfer_face;

— first_face_offset;

— second_face_offset.

4.2.22.1 chamfer_face

The chamfer_face specifies the flat transition face between the two edges of two faces. See 4.3.21 for the application assertion.

4.2.22.2 first_face_offset

The first_face_offset specifies a face and offset amount for one of the edges of the Chamfer. See 4.3.22 for the application assertion.

4.2.22.3 second_face_offset

The second_face_offset specifies a face and either an offset amount or angle amount for one of the edges of the Chamfer. See 4.3.23 for the application assertion.

4.2.23 Chamfer_angle

A Chamfer_angle is a type of Second_chamfer_offset (see 4.2.215) that is the measured angle from the Second_offset face to the Chamfer face for creating a Chamfer feature.

NOTE Figure 15 illustrates a Chamfer, a Chamfer_angle, and the angle_amount associated data.

The data associated with a Chamfer_angle are the following:

— angle_amount.

4.2.23.1 angle_amount

The Angle_amount specifies the angular measurement from a face for creating a Chamfer feature. See 4.3.24 for the application assertion.

4.2.24 Circular_boss

A Circular_boss is a type of Boss (see 4.2.11) that is a cylindrical shape. A Circular_boss may be tapered.

NOTE Figure 16 illustrates a Circular_boss with a planar top and a fillet radius, and Figure 17 illustrates a Circular_boss with taper.

The data associated with a Circular_boss are the following:

— change_in_diameter;

— circular_profile.

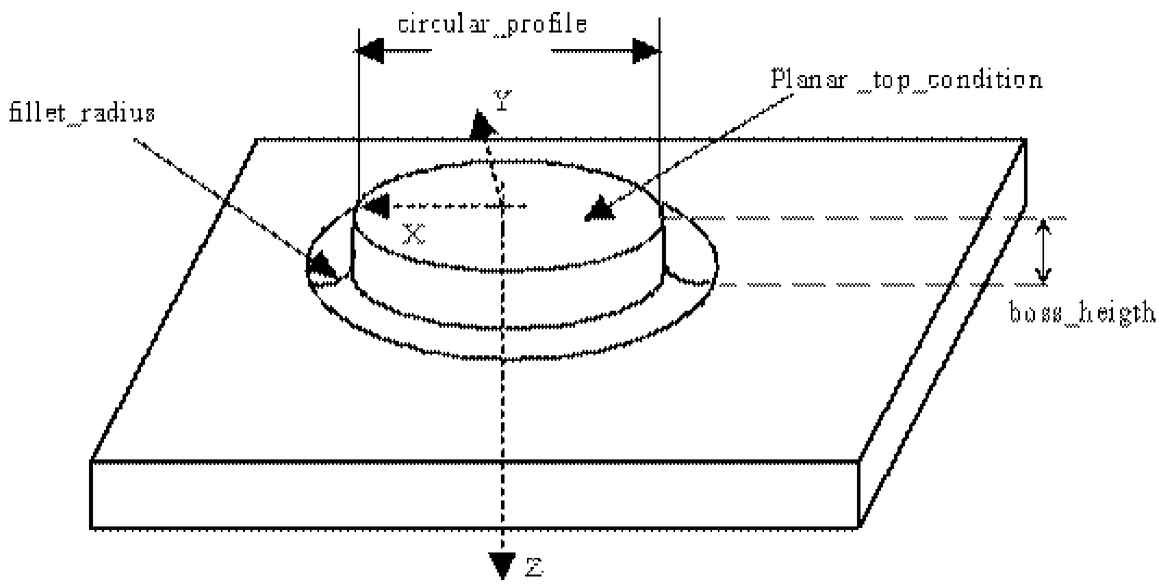


Figure 16 — Circular_boss

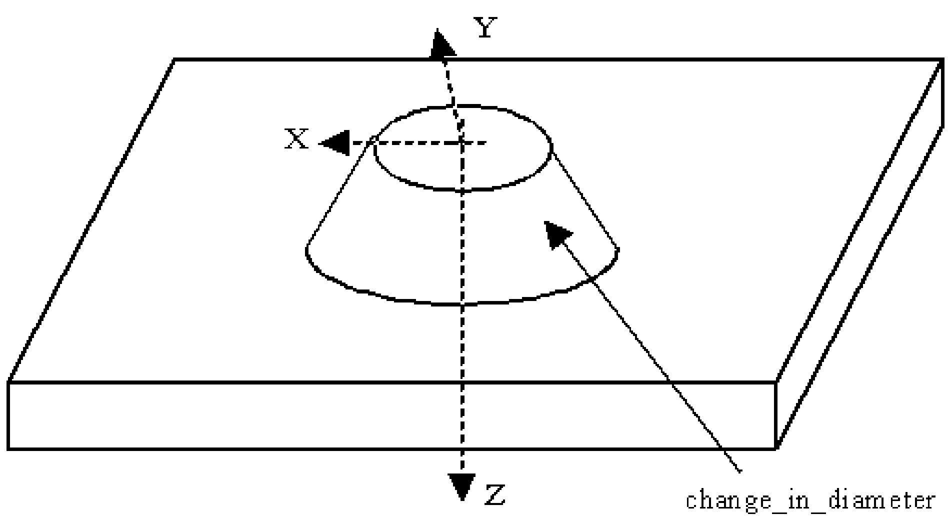


Figure 17 — Circular_boss with taper

4.2.24.1 change_in_diameter

The change_in_diameter specifies a taper that defines the change in shape of the Circular_boss. The change_in_diameter need not be specified for a particular Circular_boss. See 4.3.26, 4.3.27 and 4.3.28 for the application assertion.

4.2.24.2 circular_profile

The circular_profile specifies the diameter required by a Circular_boss. The diameter is the distance across the Circular_boss. The placement of the circular_profile shall be with the origin of the Circular_-closed_profile at the origin of the Circular_boss. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Circular_boss. See 4.3.25 for the application assertion.

4.2.25 Circular_closed_profile

A Circular_closed_profile is a type of Closed_profile (see 4.2.35) that is an enclosed area bounded by a circle. The orientation is at the center of the circle.

NOTE Figure 18 illustrates a Circular_closed_profile.

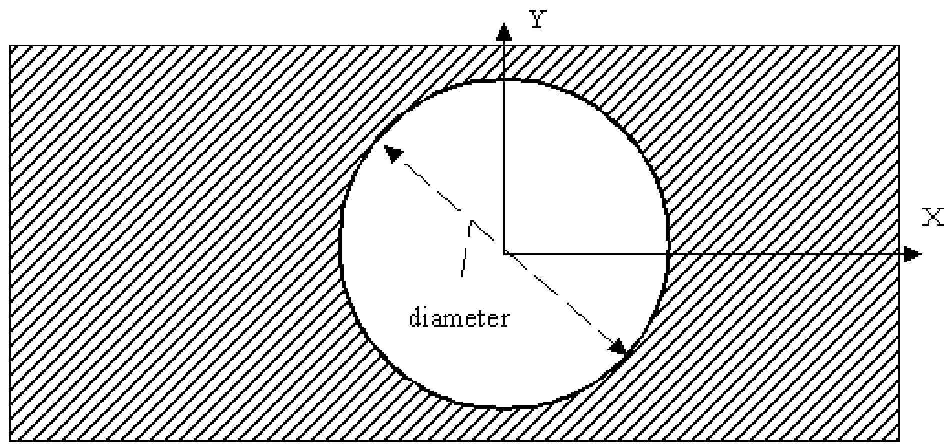


Figure 18 — Circular_closed_profile

The data associated with a Circular_closed_profile are the following:

— diameter.

4.2.25.1 diameter

The diameter specifies the distance across the Circular_closed_profile. See 4.3.29 for the application assertion.

4.2.26 Circular_closed_shape_profile

A Circular_closed_shape_profile is a type of Shape_profile (see 4.2.220) that defines a completely enclosed volume. The data associated with a Circular_closed_shape_profile are the following:

— closed_boundary.

4.2.26.1 closed_boundary

The closed_boundary specifies the outline of the Shape_profile feature. The outline defines an area that shall be enclosed and circular. The placement of the closed_profile shall be with the origin of the Path, that defines the profile, at the origin of the Circular_closed_shape_profile. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Circular_closed_shape_profile. See 4.3.30 for the application assertions.

4.2.27 Circular_cutout

A Circular_cutout is a type of Cutout (see 4.2.50) that is an enclosed volume of circular shape. A Circular_cutout is similar in definition to a Hole (see 4.2.111), but differ in the type of process required to manufacture. The data associated with a Circular_cutout are the following:

— circular_boundary.

4.2.27.1 circular_boundary

The circular_boundary is the distance across the Circular_cutout. The placement and orientation of the Circular_closed_profile (see 4.2.25) shall be the same as the Circular_cutout feature. See 4.3.31 for the application assertion.

4.2.28 Circular_offset_pattern

A Circular_offset_pattern is a Circular_pattern with a modification of the placement of a particular occurrence of the base feature relative to its expected placement.

NOTE Figure 19 illustrates a Circular_offset_pattern. The rectangular base shape is offset from index number seven.

The data associated with a Circular_offset_pattern are the following:

— angular_offset;

— index_number.

4.2.28.1 angular_offset

The angular_offset is the amount of offset from the expected feature location in a Circular_pattern, along the base feature diameter for placing another feature. See 4.3.32 for the application assertion.

4.2.28.2 index_number

The index_number specifies the value for unique identification of a location of a base feature used in a Circular_pattern. See 4.3.32 for the application assertion.

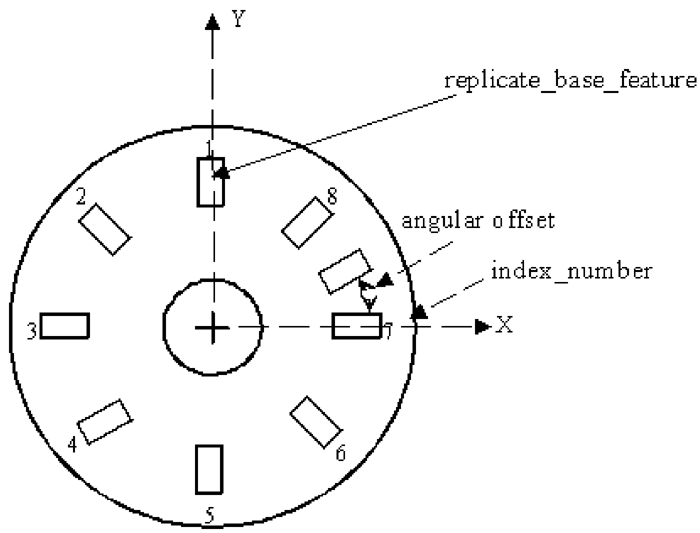


Figure 19 — Circular_offset_pattern

4.2.29 Circular_omit_pattern

A Circular_omit_pattern is a Circular_pattern with an omission of a particular occurrence of the base feature.

NOTE Figure 20 illustrates a Circular_omit_pattern with the omission of the rectangular base shape from index number 7.

The data associated with a Circular_omit_pattern are the following:

— omit_index.

4.2.29.1 omit_index

The omit_index specifies the value for unique identification of the base feature to be omitted in a Circular_pattern. See 4.3.33 for the application assertion.

4.2.30 Circular_path

A Circular_path is a type of Path (see 4.2.163) that is a direction of travel along an arc of constant radius around the Z-axis of the feature. Each Circular_path is either a Complete_circular_path (see 4.2.37) or a Partial_circular_path (see 4.2.160). The data associated with a Circular_path are the following:

— radius.

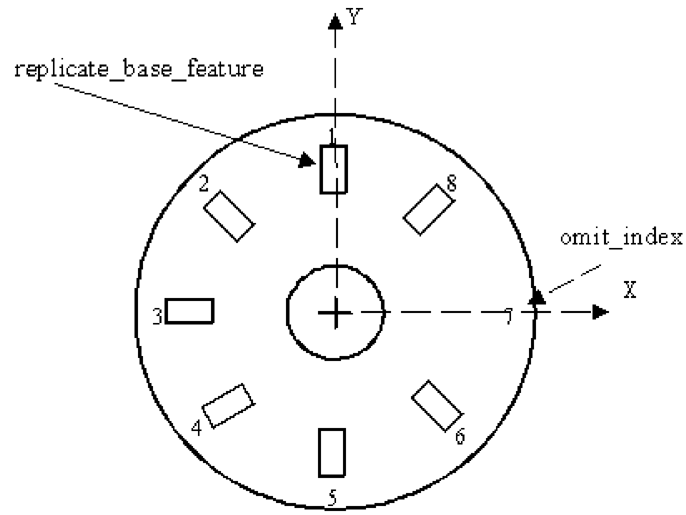


Figure 20 — Circular_omit_pattern

4.2.30.1 radius

The radius specifies the constant distance from an axis for the Circular_path. See 4.3.34 for the application assertion.

4.2.31 Circular_pattern

A Circular_pattern is a type of Replicate_feature (see 4.2.204) that is a base feature arranged in a pattern around a circular arc, equally spaced about an axis. Instances of base features need not be rotated. When a base feature rotation is required, the first feature instance is not rotated. The second through the Nth feature instance are rotated by the same angular amount measured from the placement and orientation of the preceding feature instance. (This does not imply the use of chain dimensioning.)

NOTE Figure 21 illustrates a Circular_pattern with rotation of a base feature, and Figure 22 illustrates a Circular_pattern without rotation.

The data associated with a Circular_pattern are the following:

— angular_spacing;

— base_feature_diameter;

— base_feature_rotation;

— missing_base_feature;

— number_of_features;

— relocated_base_feature.

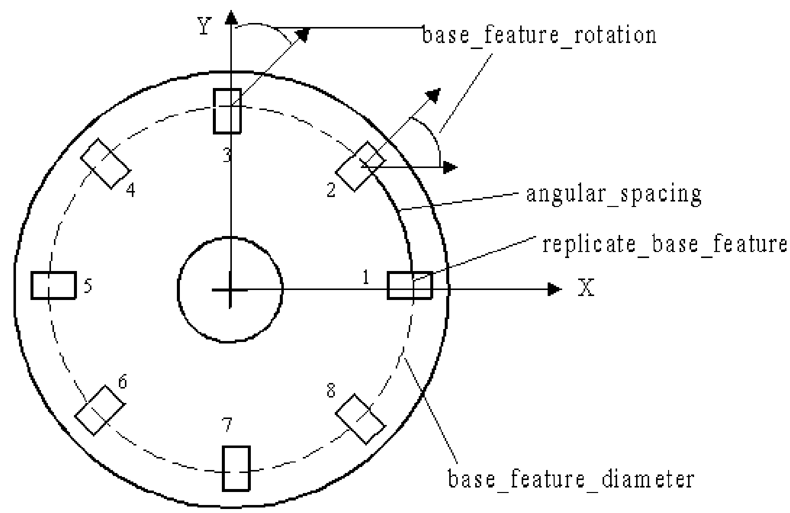


Figure 21 — Circular_pattern with rotation

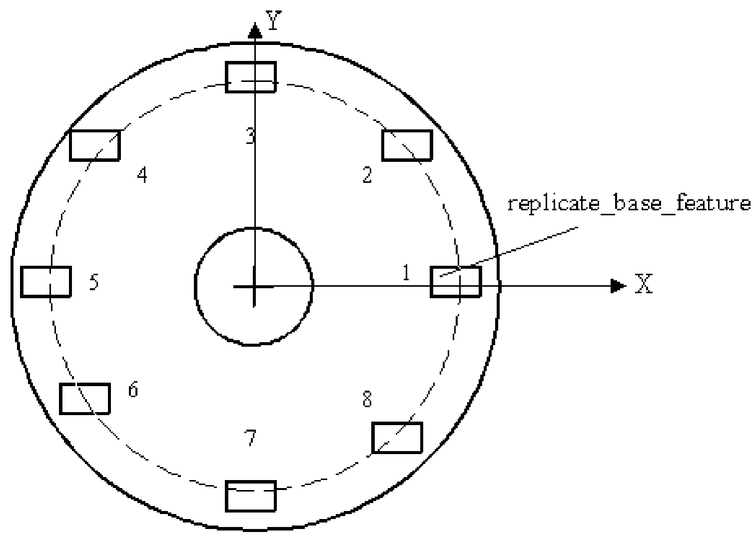


Figure 22 — Circular_pattern without rotation

4.2.31.1 angular_spacing

The angular_spacing specifies the angle amount between features in a Circular_pattern. See 4.3.37 for the application assertion.

4.2.31.2 base_feature_diameter

The base_feature_diameter specifies the size of the circle for placement of features in a Circular_pattern. See 4.3.37 for the application assertion.

4.2.31.3 base_feature_rotation

The base_feature_rotation specifies the angle to rotate one feature in regard to the orientation of a previous feature in a Circular_pattern. See 4.3.37 for the application assertion.

4.2.31.4 missing_base_feature

The missing_base_feature specifies the definition to remove any number of base features from the Circular_pattern. The missing_base_feature need not be specified for a particular Circular_pattern. There may be more than one missing_base_feature for a Circular_pattern. See 4.3.36 for the application assertion.

4.2.31.5 number_of_features

The number_of_features specifies the quantity of a base feature to be used in a Circular_pattern. See 4.3.37 for the application assertion.

4.2.31.6 relocated_base_feature

The relocated_base_feature specifies the definition to offset any number of base features from the Circular_pattern. The relocated_base_feature need not be specified for a particular Circular_pattern. There may be more than one relocated_base_feature for a Circular_pattern. See 4.3.35 for the application assertion.

4.2.32 Circular_runout_tolerance

A Circular_runout_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a tolerance to control circular elements of a part to a datum axis the part is rotated about. The tolerance is applied independent of any circular position as the part is rotated 360 degrees. Where applied to surfaces around a datum axis, Circular_runout_tolerance may be used to control the cumulative variations of circularity and coaxiality. Where applied to surfaces at right angles to the datum axis, it controls circular elements of a plane surface.

NOTE Figure 23 illustrates a Circular_runout_tolerance.

The data associated with a Circular_runout_tolerance are the following:

— geometric_reference;

— runout_angle.

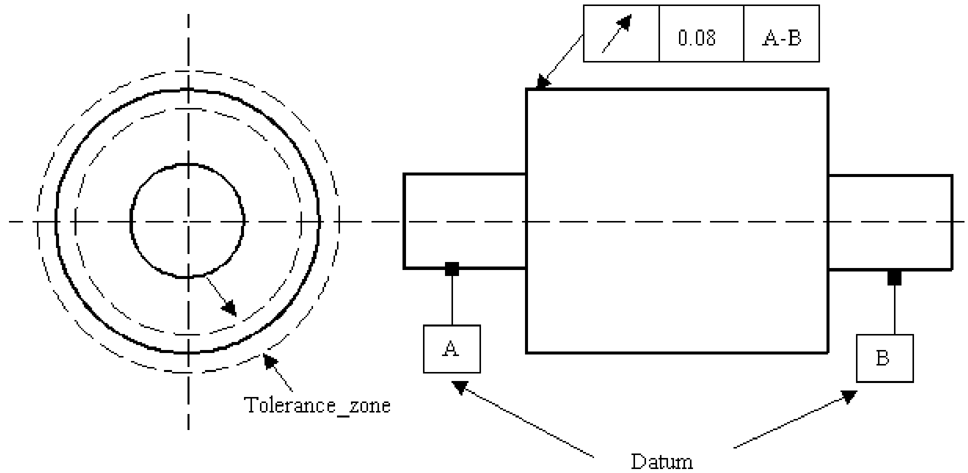


Figure 23 — Circular_runout_tolerance

4.2.32.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.38 for the application assertion.

4.2.32.2 runout_angle

The runout_angle specifies the direction to control a runout tolerance. If the angle is specified, the runout tolerance applies in this angle which is fixed with respect to the datum axis. The runout need not be specified for a particular Circular_runout_tolerance.

NOTE The Circular_runout_tolerance definition is derived from ISO 1101:2004, 18.16.

4.2.33 Circularity_tolerance

A Circularity_tolerance is a type of Geometric_tolerance (see 4.2.104) that describes the allowable deviation of a surface from round. The actual surface shall lie within a tolerance zone defined by two concentric circles.

NOTE 1 Figure 24 illustrates Circularity_tolerance.

NOTE 2 The Circularity_tolerance definition is derived from ISO 1101:2004, 18.13.

NOTE 3 In ISO 1101:2004. “circularity tolerance” is referred to as “roundness tolerance”.

4.2.34 Class_BSU

A Class_BSU is a type of BSU (see 4.2.17) that identifies a class in a parts library. The combination of supplier identification, code, and version of a class shall be unique.

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

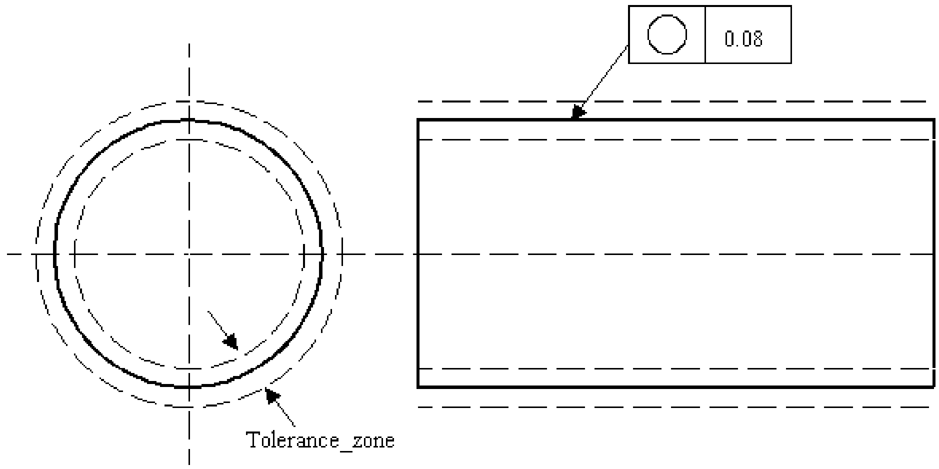


Figure 24 — Circularity_tolerance

The data associated with a Class_BSU are the following:

— defined_by;

— version.

4.2.34.1 defined_by

The defined_by specifies the library supplier who defines the library class. See 4.3.39 for the application assertion.

4.2.34.2 version

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

4.2.35 Closed_profile

A Closed_profile is a type of Profile (see 4.2.181) that is an outline or shape that bounds an enclosed area with no opening. Each Closed_profile is either a Circular_closed_profile (see 4.2.25), a General_closed_-profile (see 4.2.90), a Ngon_profile (see 4.2.142), or a Rectangular_closed_profile (see 4.2.196).

4.2.36 Closed_slot

A Closed_slot is a type of Slot (see 4.2.224) that has a course of travel defined by a Path that is closed and has no opening. The data associated with a Closed_slot are the following:

— course_of_travel;

— end_condition.

4.2.36.1 course_of_travel

The course_of_travel specifies a 3D space curve, that when combine with a Profile, creates the shape of the Slot. See 4.3.40 and 4.3.41 for the application assertion.

4.2.36.2 end_condition

The end_conditions specifies the type of closed shape at the ends of the Slot. See 4.3.42 for the application assertion.

4.2.37 Complete_circular_path

A Complete_circular_path is a type of Circular_path (see 4.2.30) that is a direction of travel that begins and ends at the same point on the arc.

NOTE Figure 25 illustrates a Slot feature with a Square_u_profile and a Complete_circular_path.

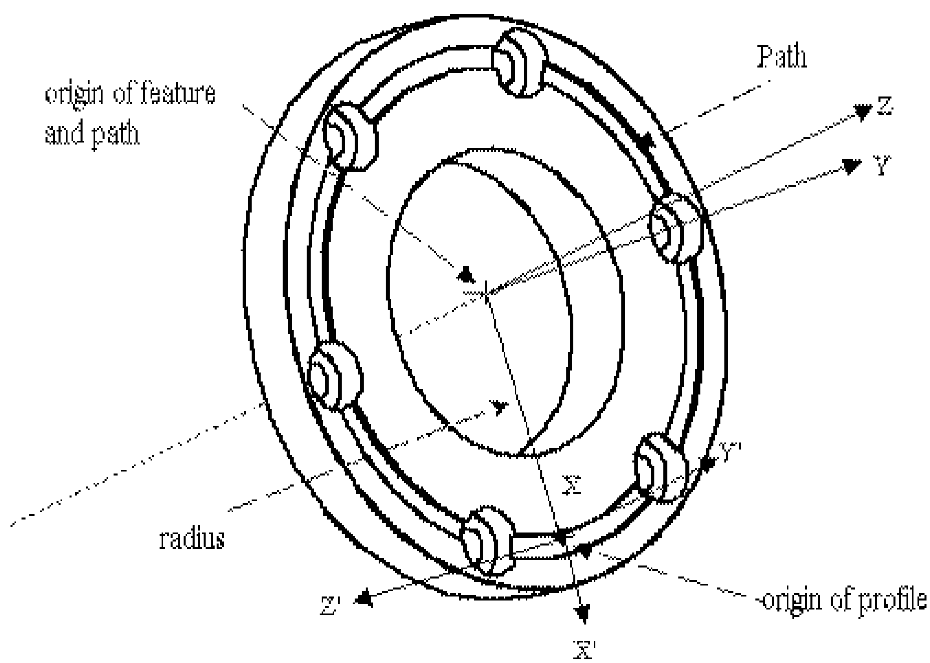


Figure 25 — Complete_circular_path

4.2.38 Compound_datum

A Compound_datum is a type of Datum (see 4.2.54) that is a set of two or more Datum_feature objects which are for establishing a single datum plane or axis. The data associated with a Compound_datum are the following:

— element.

4.2.38.1 element

The element specifies the list of Datum_feature objects for defining the Compound_datum. See 4.3.43 for the application assertion.

4.2.39 Compound_feature

A Compound_feature is a type of Machining_feature (see 4.2.126) that is a union of one or more Machining_feature objects to create a more complex feature definition. The placement of a Compound_-feature is relative to either the part, another Compound_feature, or a Replicate_feature which uses a Compound_feature as the base feature. Features which are elements of the Compound_feature have placement defined relative to the Compound_feature placement.

NOTE Figure 26 illustrates an example of a Compound_feature with one counterbore_hole feature and one countersunk_hole feature.

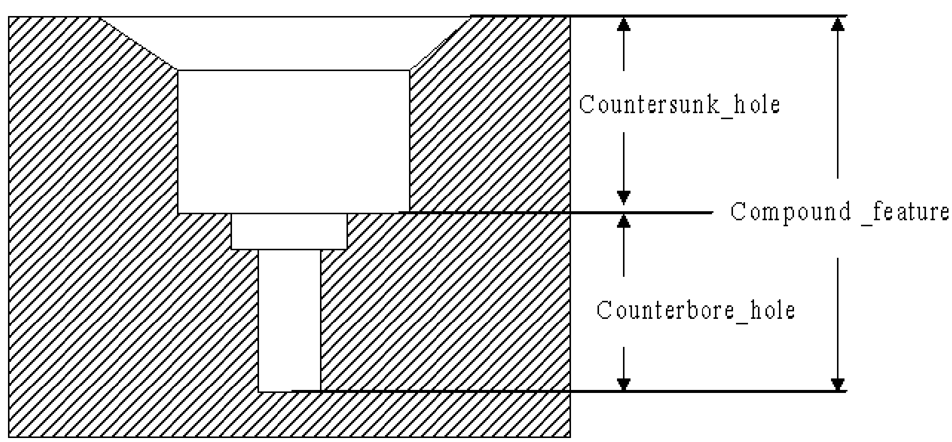


Figure 26 — Compound_feature

The data associated with a Compound_feature are the following:

— element;

— feature_description;

— feature_name.

4.2.39.1 element

The element specifies the base features that when combined defines a Compound_feature. There shall be more than one element for a Compound_feature. See 4.3.44 for the application assertion.

4.2.39.2 feature_description

The feature_description specifies a user defined explanation of the Compound_feature. A Compound_-feature may but need not require a feature_description.

4.2.39.3 feature_name

The feature_name specifies a user defined name for the Compound_feature that need not be unique.

4.2.40 Compound_feature_element

A Compound_feature_element specifies the type of Machining_feature (see 4.2.126) or Transition_-feature (see 4.2.258) to be used for a particular element of a Compound_feature. A Compound_feature consists of two or more Compound_feature_element objects and need not require Compound_feature_-element objects to be ordered. The data associated with a Compound_feature_element are the following:

— element.

4.2.40.1 element

The element specifies the base feature to be used as one of the components for the Compound_feature. See 4.3.45 and 4.3.46 for the application assertions.

4.2.41 Compound_feature_relationship

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

The data associated with a Compound_feature_relationship are the following:

— predecessor;

— successor.

4.2.41.1 predecessor

The predecessor specifies the Compound_feature_element with the highest precedence. See 4.3.47 for the application assertion.

4.2.41.2 successor

The successor specifies Compound_feature_element with a lesser precedence. See 4.3.47 for the application assertion.

4.2.42 Concentricity_tolerance

A Concentricity_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a cylindrical or conical feature of a part which shall be fundamentally concentric. When the part is rotated about the datum axis, the axis of the feature shall be within the cylindrical tolerance zone.

NOTE 1 Figure 27 illustrates the Concentricity_tolerance

NOTE 2 The Concentricity_tolerance definition is derived from ISO 1101:2004, 18.13.

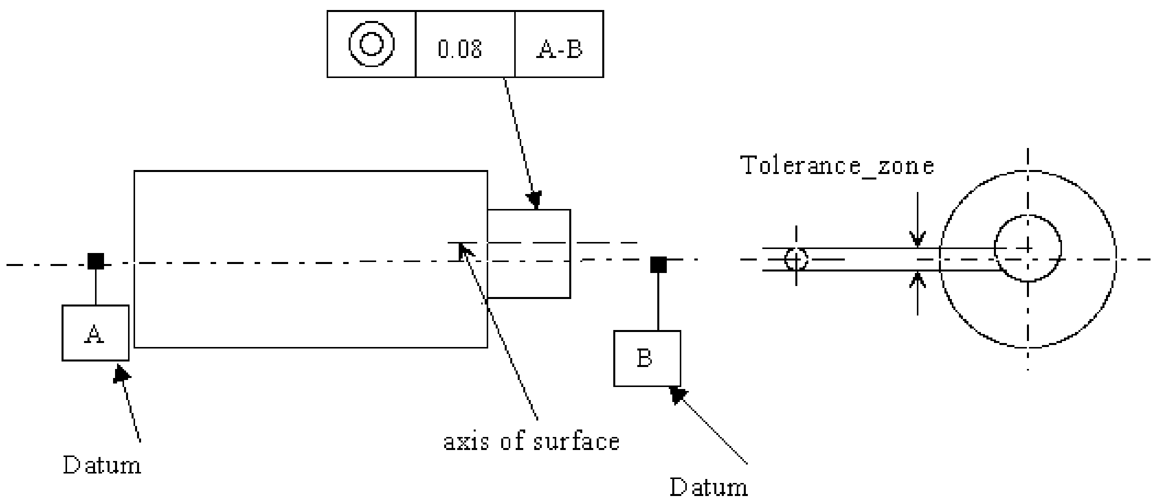


Figure 27 — Concentricity_tolerance

The data associated with a Concentricity_tolerance are the following:

— geometric_reference;

— value_qualifier.

4.2.42.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.48 for the application assertion.

4.2.42.2 value_qualifier

The value_qualifier specifies whether the Concentricity_tolerance is either a diametric tolerance or is not.

4.2.43 Conical_hole_bottom

A Conical_hole_bottom is a type of Blind_bottom_condition (see 4.2.9) that defines the bottom of a Round_hole to be conical in shape. A Conical_hole_bottom shall be a constant decrease in the hole diameter until the radius is zero. The Conical_hole_bottom may have a tip_radius defined at the smallest end of the Conical_hole_bottom.

NOTE Figure 28 illustrates the Conical_hole_bottom for a Round_hole.

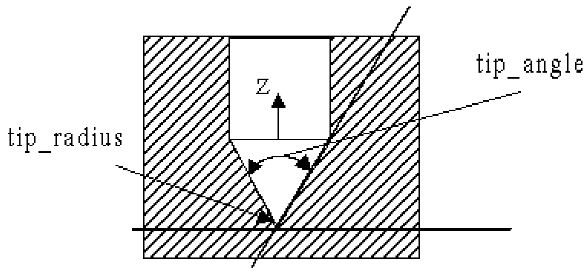


Figure 28 — Conical_hole_bottom

The data associated with a Conical_hole_bottom are the following:

— tip_angle;

— tip_radius.

4.2.43.1 tip_angle

The tip_angle specifies the amount of constant slope to decrease the Round_hole diameter until the diameter is zero. The tip_angle is a conical bottom for a Round_hole. The tip_angle shall be greater than 0 degrees and less than 180 degrees. See 4.3.49 for the application assertion.

4.2.43.2 tip_radius

The tip_radius specifies the radius of a conical bottom for a Round_hole. A conical bottom is a constant decrease in diameter until the diameter is zero, or until it becomes tangent to a tip_radius. The tip_radius need not be specified for a particular Conical_hole_bottom. See 4.3.49 for the application assertion.

4.2.44 Constant_radius_edge_round

A Constant_radius_edge_round is a type of Edge_round (see 4.2.73) that is defined with a constant radius value.

NOTE Figure 29 illustrates the Constant_radius_edge_round on a rectangular block.

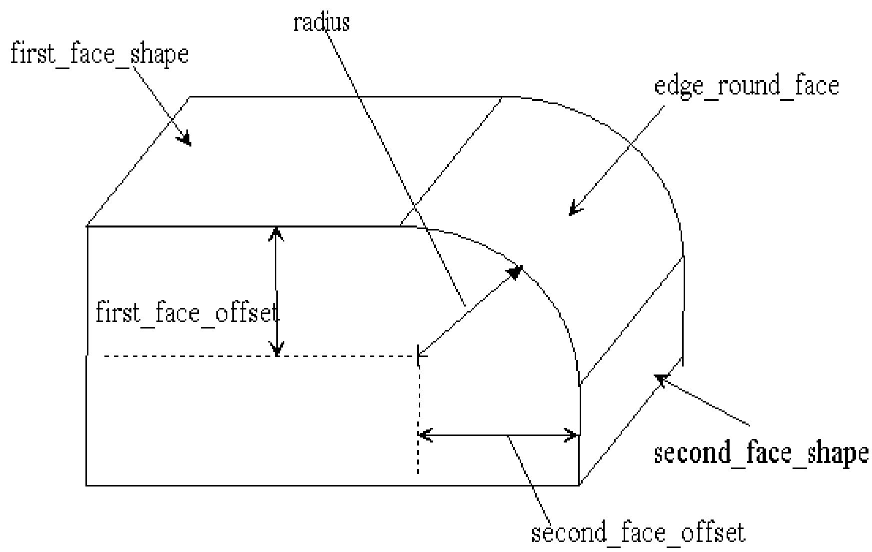


Figure 29 — Constant_radius_edge_round

The data associated with a Constant_radius_edge_round are the following:

— first_face_offset;

— second_face_offset;

— radius.

4.2.44.1 first_face_offset

The first_face_offset specifies the amount of offset from the first_face_shape (see 4.2.73.2) to the center of the edge_round_face (see 4.2.73.1). A constant_radius_edge_round may but need not require a first_-face_offset. See 4.3.50 for the application assertion.

4.2.44.2 second_face_offset

The second_face_offset specifies the amount of offset from the second_face_shape (see 4.2.73.3) to the center of the edge_round_face (see 4.2.73.1). A constant_radius_edge_round may but need not require a second_face_offset. See 4.3.50 for the application assertion.

4.2.44.3 radius

The radius specifies the amount of curvature for a convex transition between the two faces of a Constant_radius_edge_round. See 4.3.50 for the application assertion.

4.2.45 Constant_radius_fillet

A Constant_radius_fillet is a type of Fillet (see 4.2.81) that is defined with a constant radius value.

NOTE Figure 30 illustrates the Constant_radius_fillet. The Fillet surface is not necessarily tangent to the other surfaces because the offsets.

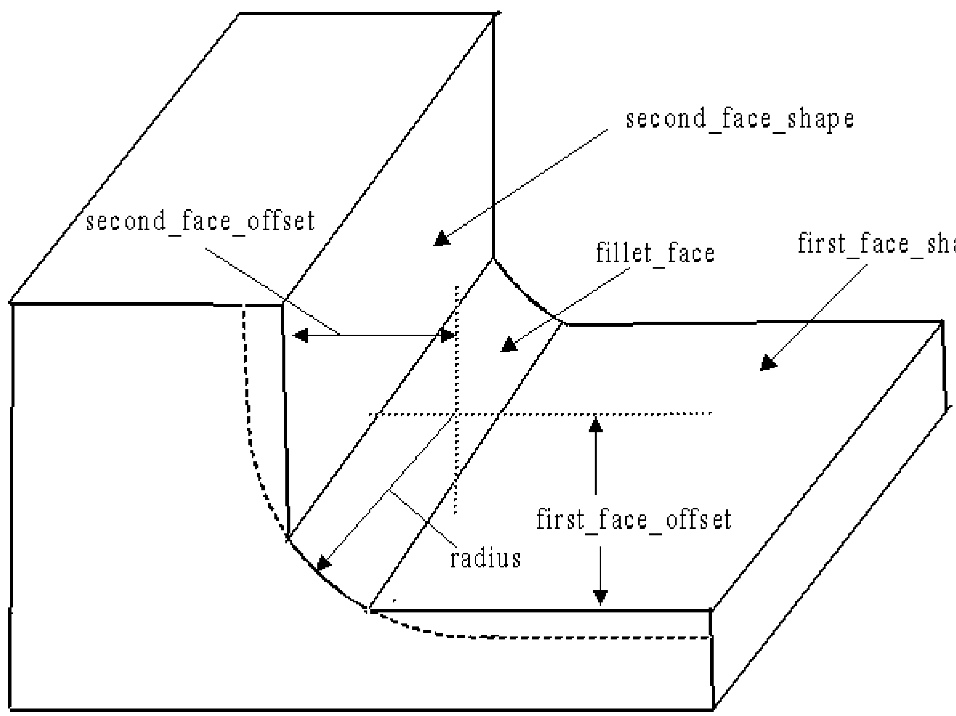


Figure 30 — Constant_radius_fillet

The data associated with a Constant_radius_fillet are the following:

— first_face_offset;

— second_face_offset;

— radius.

4.2.45.1 first_face_offset

The first_face_offset specifies the amount of offset from the first_face_shape (see 4.2.81.2) to the center of the fillet_face (see 4.2.81.1). A constant_radius_fillet may but need not require a first_face_offset. See 4.3.51 for the application assertion.

4.2.45.2 second_face_offset

The second_face_offset specifies the amount of offset from the second_face_shape (see 4.2.81.3) to the center of the fillet_face (4.2.81.1). A constant_radius_fillet may but need not require a first_face_offset. See 4.3.51 for the application assertion.

4.2.45.3 radius

The radius specifies the amount of curvature for a concave transition between two surfaces. See 4.3.51 for the application assertion.

4.2.46 Counterbore_hole

A Counterbore_hole is a type of Hole (see 4.2.111) that is a combination of two Round_holes. The first Round_hole shall have either a Through_bottom_condition or Blind_bottom_condition, the second shall have a Blind_bottom_condition, and a larger diameter than the first Round_hole. The top of the first Round_hole shall mate with the bottom of the second Round_hole. The Counterbore_hole orientation shall be the same as the orientation of the first Round_hole. Both Round_holes shall be co-axial.

NOTE Figure 31 illustrates the Counterbore_hole with a Blind_bottom_condition and a Through_bottom_condition.

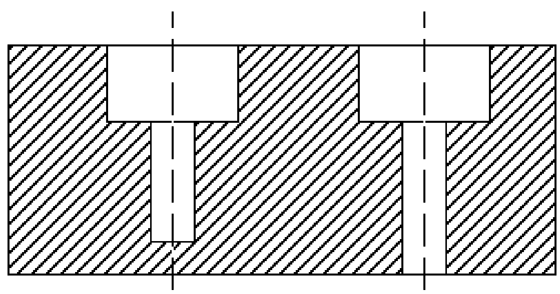


Figure 31 — Counterbore_hole

The data associated with a Counterbore_hole are the following:

— larger_hole;

— smaller_hole.

4.2.46.1 larger_hole

The larger_hole specifies the Round_hole that will be used as the larger hole for the Counterbore_hole. See 4.3.52 for the application assertion.

4.2.46.2 smaller_hole

The smaller_hole specifies the Round_hole that will be used as the smaller hole for the Counterbore_hole. See 4.3.52 for the application assertion.

4.2.47 Countersunk_hole

A Countersunk_hole is a type of Hole (see 4.2.111) that is a combination of two Round_holes. The first Round_hole shall have a Through_bottom_condition or Blind_bottom_condition. The second shall be a Round_hole with a Blind_bottom_condition, and a taper. The top of the first Round_hole shall mate with the bottom of the second Round_hole. The taper of the second Round_hole shall be larger than the diameter of the Round_hole, decreasing to the same diameter at the point where the two holes join. The Countersunk_hole orientation shall be the same as the orientation of the first Round_hole. Both Round_holes shall be co-axial.

NOTE Figure 32 illustrates the Countersunk_hole with a Blind_bottom_condition and a Through_bottom_condition.

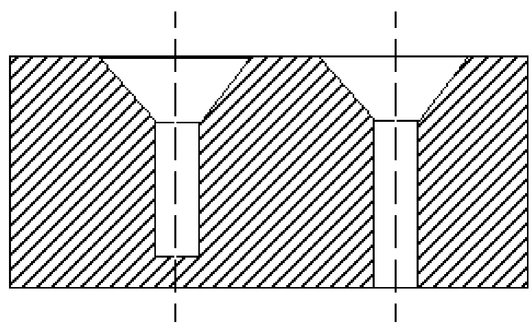


Figure 32 — Countersunk_hole

The data associated with a Countersunk_hole are the following:

— constant_diameter_hole;

— tapered_hole.

4.2.47.1 constant_diameter_hole

The constant_diameter specifies the Round_hole without a taper that will be used as the constant diameter hole for the Countersunk_hole. See 4.3.53 for the application assertion.

4.2.47.2 tapered_hole

The tapered_hole specifies the Round_hole with a taper that will be used as the tapered hole for the Counterbore_sunk. See 4.3.53 for the application assertion.

4.2.48 Curved_dimension_tolerance

A Curved_dimension_tolerance is a type of Size_tolerance (see 4.2.223) that is the tolerance on a dimension for a curve measured along the entire path of the curve.

NOTE Figure 33 illustrates the Curved_dimension_tolerance.

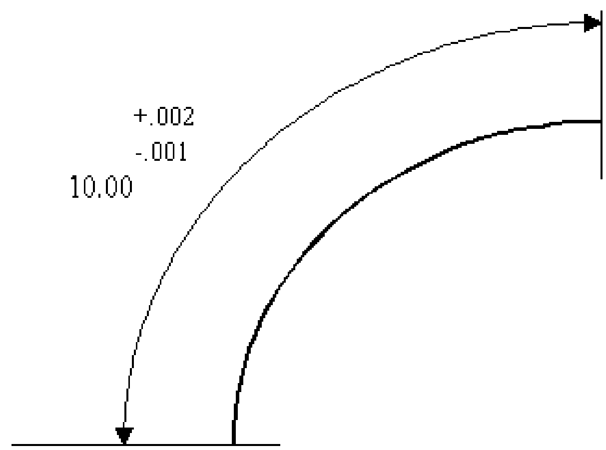


Figure 33 — Curved_dimension_tolerance

4.2.49 Customer_order

A Customer_order is a document, received from a customer, that describes data to process a request for the manufacture of a part. The data associated with a Customer_order are the following:

— customer;

— delivery_date;

— initiated_order;

— material_disposition;

— order_number;

— order_status;

— quantity_ordered;

— special_instructions.

4.2.49.1 customer

The customer specifies the person and company requesting the manufacture of the part. See 4.3.55 for the application assertion.

4.2.49.2 delivery_date

The delivery_date specifies the year, month and day when the part is to be received by the customer.

4.2.49.3 initiated_order

The initiated_order specifies Project_order that is initiated by the Customer_order. There may be more than one initiated_order for a Customer_order. See 4.3.56 for the application assertion.

4.2.49.4 material_disposition

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

4.2.49.5 order_number

The order_number specifies a unique identification of the Customer_order.

4.2.49.6 order_status

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

4.2.49.7 quantity_ordered

The quantity_ordered specifies number of parts to be manufactured (see Ordered_part 4.2.148). There may be more than one quantity_ordered for a Customer_order. See 4.3.54 for the application assertion.

4.2.49.8 special_instructions

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

4.2.50 Cutout

A Cutout is a type of Pocket (see 4.2.177) that is a volume to be removed from the part. Cutouts shall pass through two faces of a Part. Each Cutout is either a Circular_cutout (see 4.2.27)or a General_cutout (see 4.2.91).

NOTE Figure 34 illustrates the Cutout.

The data associated with a Cutout are the following:

— bottom_condition.

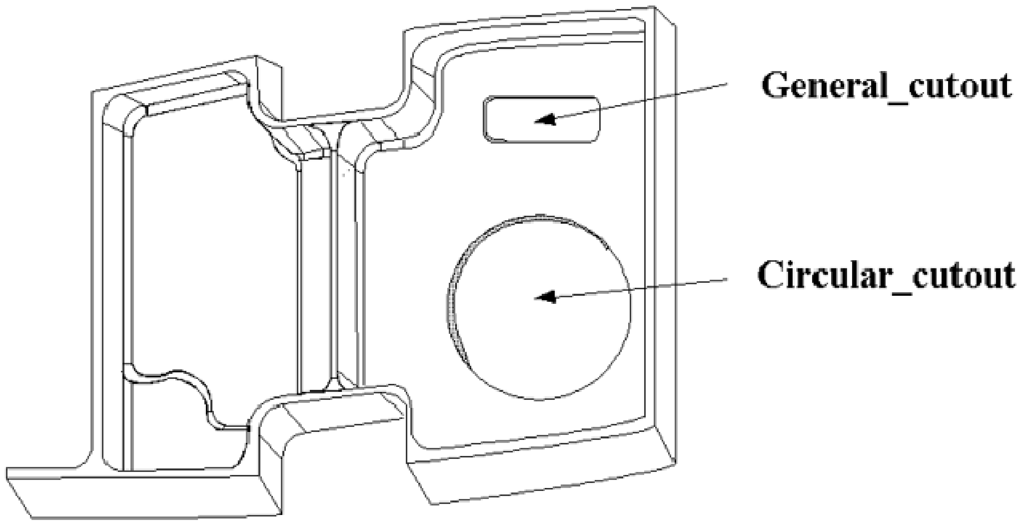


Figure 34 — Cutout

4.2.50.1 bottom_condition

The bottom_condition specifies the shape of the bottom of a Cutout feature. The bottom_condition shall pass entirely through the part, it shall be a Through_pocket_bottom_condition (see 4.2.250). See 4.3.57 for the application assertion.

4.2.51 Cutting_tool_requisition

A Cutting_tool_requisition is a type of Requisition (see 4.2.205) that is an organization's requirements for the purchase of tools required to support part manufacturing.

4.2.52 Cylindrical_base_shape

A Cylindrical_base_shape is a type of Implicit_base_shape_representation (see 4.2.112) that is the initial shape of the material which is cylindrical.

NOTE Figure 35 illustrates a Cylindrical_base_shape.

The data associated with a Cylindrical_base_shape are the following:

— diameter.

4.2.52.1 diameter

The diameter specifies the distance across a Cylindrical_base_shape. See 4.3.58 for the application assertion.

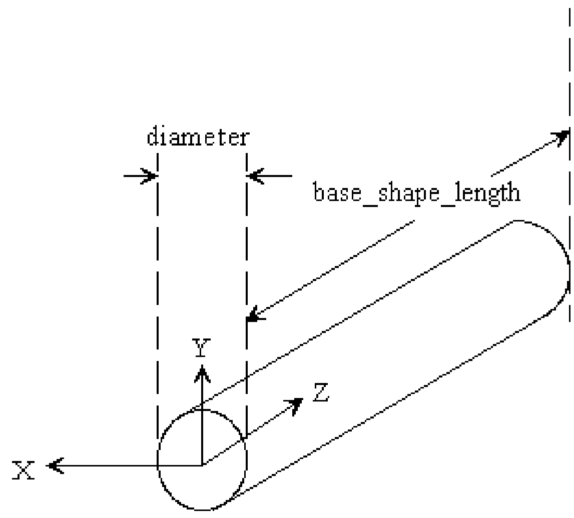


Figure 35 — Cylindrical_base_shape

4.2.53 Cylindricity_tolerance

A Cylindricity_tolerance is a type of Geometric_tolerance (see 4.2.104) that describes the amount of deviation a feature may have from being truly cylindrical. The feature shall be contained between two coaxial cylinders. The distance between the two coaxial cylinders defines the allowable tolerance deviation.

NOTE 1 Figure 36 illustrates Cylindricity_tolerance.

NOTE 2 The Cylindricity_tolerance definition is derived from ISO 1101:2004, 18.4.

4.2.54 Datum

A Datum is defined in 3.1 of ISO 5459. A Datum is either a Datum_feature or a Compound_datum. The data associated with a Datum are the following:

— name;

— precedence.

4.2.54.1 name

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

4.2.54.2 precedence

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

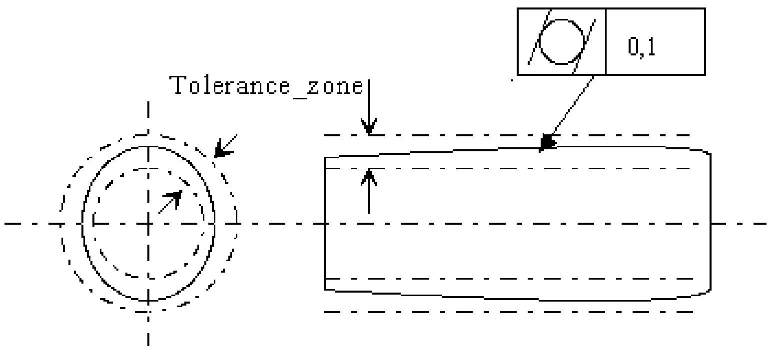


Figure 36 — Cylindricity_tolerance

4.2.55 Datum_feature

A Datum_feature is a type of Datum (see 4.2.54) that is a feature on a part used to establish a Datum. The data associated with a Datum_feature are the following:

— datum_representation;

— modifier.

4.2.55.1 datum_representation

The datum_representation specifies the type of representation for defining a Datum_feature. The Datum_feature may be defined by either a Datum_target definition or by Part shape representation. See 4.3.60 and 4.3.61 for the application assertions.

4.2.55.2 modifier

The modifier specifies the tolerance value applied to the Datum_feature. The modifier need not be specified for a particular Datum_feature. See 4.3.59 for the application assertion.

4.2.56 Datum_target

A Datum_target is a geometric element on the surface of a part to locate a Datum for reference by a Geometric_tolerance. A Datum_target is either a Placed_target (see 4.2.169) or a Target_area (see 4.2.240). The data associated with a Datum_target are the following:

— identifier.

4.2.56.1 identifier

The identifier specifies a unique identification for the Datum_target.

4.2.57 Datum_target_set

A Datum_target_set is a set of Datum_target objects that are used to define a datum reference for a Geometric_tolerance. The data associated with a Datum_target_set are the following:

— rule_description;

— target_shape.

4.2.57.1 rule_description

The rule_description specifies the type of datum that is formed by the Datum_target_set The rule_description need not be specified for a particular Datum_target_set.

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

4.2.57.2 target_shape

The target_shape specifies the set of Datum targets that define the Datum_target_set. There may be more than one target_shape for a Datum_target. See 4.3.62 for the application assertion.

4.2.58 Dedicated_fixture_requisition

A Dedicated_fixture_requisition is a type of Requisition (see 4.2.205) for dedicated fixtures required to support part manufacturing.

4.2.59 Defined_gear

A Defined_gear is a type of Gear (see 4.2.88) that is specified explicitly, all attributes for a Gear are declared and defined. Bevel_gear(see 4.2.8), Helical_gear(see 4.2.110) or Spur_gear (see 4.2.230).

4.2.60 Defined_marking

A Defined_marking is a type of Marking (see 4.2.131) that is specified explicitly, all attributes for a Marking are declared and defined.

NOTE Figure 37 illustrates the Defined_marking on a rectangular block shape and with text of 'TEXT FOR A MARKING'.

The data associated with a Defined_marking are the following:

— character_height;

— character_spacing;

— font_name;

— special_instructions.

4.2.60.1 character_height

The character_height specifies the size of the text used for a Defined_marking feature. See 4.3.63 for the application assertion.

4.2.60.2 character_spacing

The character_spacing specifies the amount of space between text letters used for a Defined_marking feature. See 4.3.64 for the application assertion.

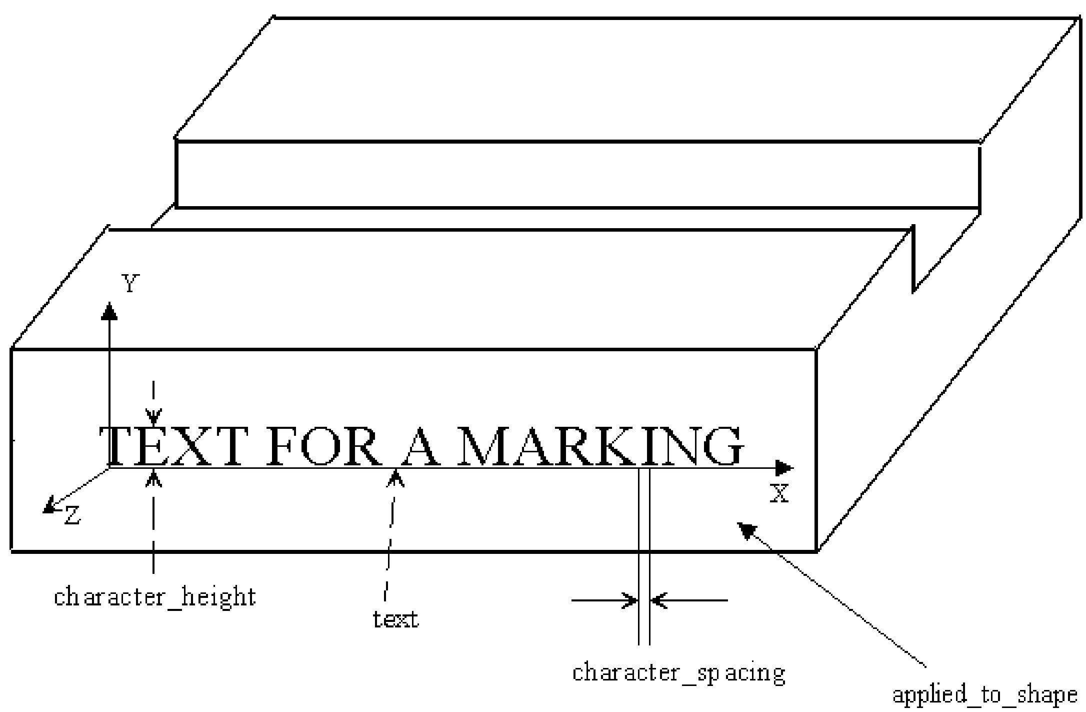


Figure 37 — Defined_marking

4.2.60.3 font_name

The font_name specifies the appearance of the characters. A font consists of typeface, treatment, and size. See 4.3.63 for the application assertion.

EXAMPLE Examples of font_name characteristic are type face Times Roman a treatment of Bold or Italic size 10 point.

4.2.60.4 special_instructions

The special_instructions specifies a description on how to apply the text given by the Defined_marking entity. See 4.3.63 for the application assertion.

4.2.61 Defined_thread

A Defined_thread is a type of Thread (see 4.2.246) that is specified explicitly.

NOTE Figure 38 illustrates a Defined_thread. The drawing note '1.500-6 UNC-2A' applies to the thread and is separated into several of the Defined_thread attributes.

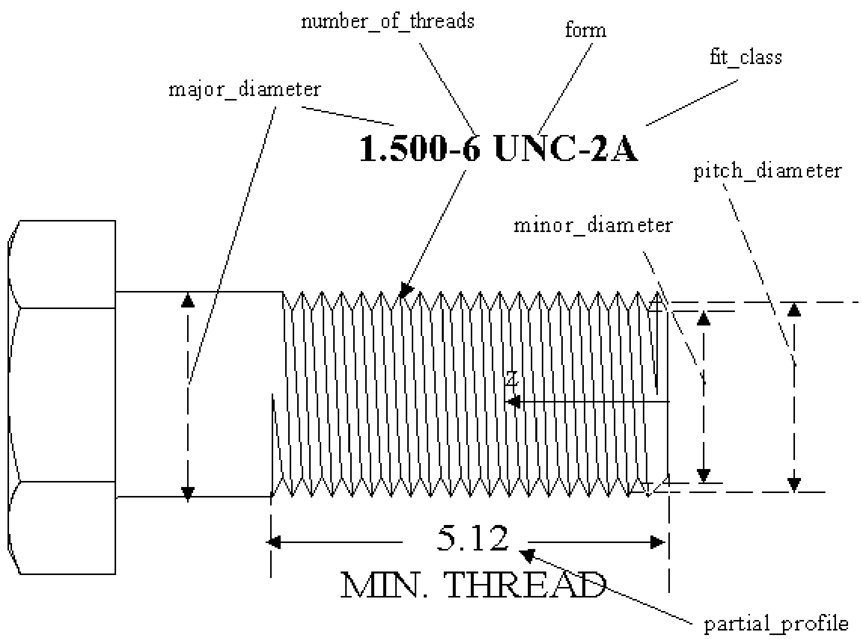


Figure 38 — Defined_thread

The data associated with a Defined_thread are the following:

— crest;

— major_diameter;

— minor_diameter;

— pitch_diameter.

4.2.61.1 crest

The crest specifies the distance between the opposing points of the thread. The crest is formed by the intersection of the sides of the thread if extended, if necessary, beyond the top of the thread. The crest need not be specified for a particular Defined_thread. See 4.3.65 for the application assertion.

4.2.61.2 major_diameter

The major_diameter specifies the dimension of the largest diameter of the Thread and is applied to both an internal and an external thread. See 4.3.65 for the application assertion.

4.2.61.3 minor_diameter

The minor_diameter specifies the dimension of the smallest diameter of the Defined_thread and is applied to both an internal and an external thread. The minor_diameter need not be specified for a particular Defined_thread. See 4.3.65 for the application assertion.

4.2.61.4 pitch_diameter

The pitch_diameter specifies the dimension of an imaginary cylinder passing through the threads so as to make equal the widths of the threads and the widths of the spaces cut by the cylinder. See 4.3.65 for the application assertion.

4.2.62 Descriptive_parameter

A Descriptive_parameter is a type of Property_parameter (see 4.2.188) that is an explanation of the property being defined by a specification. The data associated with a Descriptive_parameter are the following:

— descriptive_string.

4.2.62.1 descriptive_string

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

4.2.63 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. Process planning cannot continue until a technical recommendation is made to correct the problem. The data associated with a Design_exception_notice are the following:

— discrepant_part;

— issues;

— issuing_date;

— notice_description;

— notice_number;

— technical_recommendation.

4.2.63.1 discrepant_part

The discrepant_part specifies the set of parts that has a design discrepancy. There may be more than one discrepant_part for a Design_exception_notice. See 4.3.67 for the application assertion.

4.2.63.2 issues

The issues specifies the set of change proposals to modify a Part. The result need not be specified for a particular Design_exception_notice. There may be more than one result for a Design_exception_notice. See 4.3.66 for the application assertion.

4.2.63.3 issuing_date

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

4.2.63.4 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.

4.2.63.5 notice_number

The notice_number specifies a unique identification for each Design_exception_notice.

4.2.63.6 technical_recommendation

The technical_recommendation specifies a recommended resolution to a design problem discovered during the creation of the process plan for a part.

4.2.64 Diagonal_knurl

A Diagonal_knurl is a type of Turned_knurl (see 4.2.259) with helical cuts at an angle about the axis of a surface.

NOTE Figure 39 illustrates a Diagonal_knurl with a right hand helix.

The data associated with a Diagonal_knurl are the following:

— helix_angle;

— helix_hand.

4.2.64.1 helix_angle

The helix_angle specifies the angle the knurl pattern makes with the orientation axis of an applied to surface. See 4.3.69 for the application assertion.

4.2.64.2 helix_hand

The helix_hand specifies a description of whether the helix angle is applied to an orientation axis in a clockwise or a counterclockwise direction. See 4.3.68 for the application assertion.

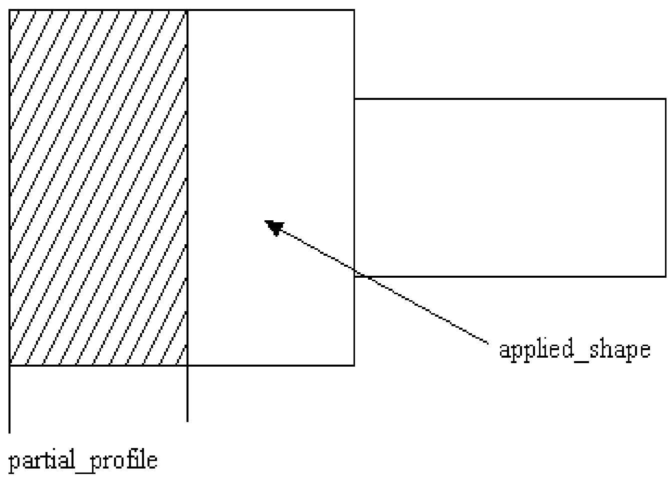


Figure 39 — Diagonal_knurl

4.2.65 Diameter_dimension_tolerance

A Diameter_dimension_tolerance is a type of Size_tolerance (see 4.2.223)that is the allowable variation of the size of a hole in a surface.

NOTE Figure 40 illustrates the Diameter_dimension_tolerance.

4.2.66 Diameter_taper

A diameter_taper is a constant change in shape of a feature for a part. A Diameter_taper starts at the placement of a feature and is applied to the entire feature. The initial diameter and the length of the taper is determined from the Machining_feature that is applying the Diameter_taper. The final_diameter is specified as a diameter different than the initial diameter.

NOTE Figure 41 illustrates the Diameter_taper.

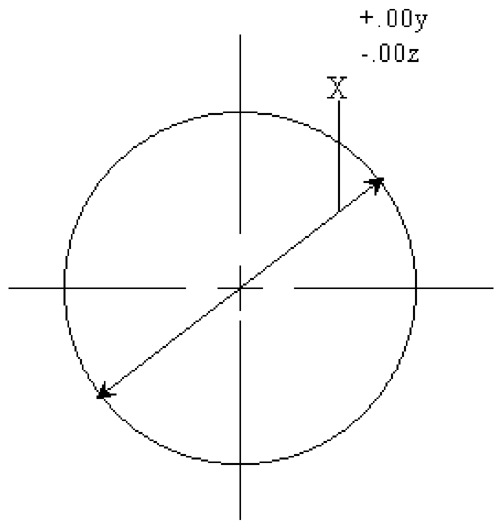


Figure 40 — Diameter_dimension_tolerance

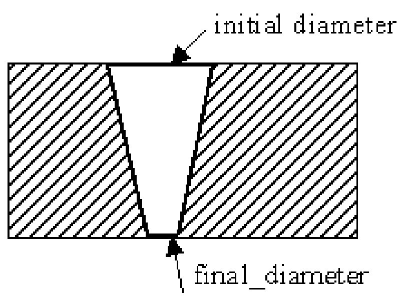


Figure 41 — Diameter_taper

The data associated with a Diameter_taper are the following:

— final_diameter.

4.2.66.1 final_diameter

The final_diameter specifies the diameter of the circle at the end of the taper. The final_diameter may be smaller or larger than the beginning diameter for a Machining_feature. See 4.3.70 for the application assertion.

NOTE The diameter at the beginning of the Diameter_taper is the same as the referencing feature diameter.

4.2.67 Diamond_knurl

A Diamond_knurl is a type of Turned_knurl (see 4.2.259) that is a knurl ridge that is doubly helical, a left hand and a right hand helix, about the axis of a surface, with equal spacing of the two.

NOTE Figure 42 illustrates a Diamond_knurl.

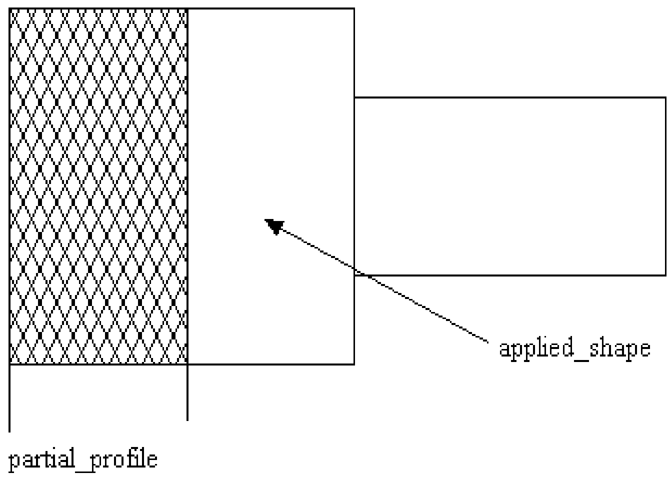


Figure 42 — Diamond_knurl

The data associated with a Diamond_knurl are the following:

— helix_angle.

4.2.67.1 helix_angle

The helix_angle specifies the angle the knurl pattern makes with the orientation axis of an applied to surface. See 4.3.71 for the application assertion.

4.2.68 Digital_technical_data_package_work_order

A Digital_technical_data_package_work_order is a document that defines the need to create and track the digital product data definition for a part. The data associated with a Digital_technical_data_package_-work_order are the following:

— order_id.

4.2.68.1 order_id

The order_id specifies a unique identifier for the Digital_technical_data_package_work_order.

4.2.69 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 4.2.123) or a Size_tolerance (see 4.2.223).

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

The data associated with a Dimensional_tolerance are the following:

— dimension_description;

— dimension_note;

— dimension_value;

— limit;

— significant_digits;

— unit_of_measure.

4.2.69.1 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.

4.2.69.2 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 4.2.69.2.1 to 4.2.69.2.3 for the definition of each allowable value for limit_qualifier.

4.2.69.2.1 auxilary: Restrict auxilary exact dimension.Dimension_tolerance to be a nominal value with no value_limitation.

4.2.69.2.2 theoretical: Restrict theoretically exact dimension.Dimension_tolerance to be a nominal value with no value_limitation.

4.2.69.2.3 user defined: a description specified by the user.

4.2.69.3 dimension_value

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

4.2.69.4 limit

The limit specifies the tolerance value applied to the Dimension_tolerance. The limit need not be specified for a particular Dimensional_tolerance. See 4.3.72 for the application assertion.

4.2.69.5 significant_digits

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

4.2.69.6 unit_of_measure

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

4.2.70 Directed_taper

A Directed_taper is a constant change in shape of a feature for a part. A Directed_taper starts at the location of the placement of a feature and is applied to the entire feature. The length of the taper is determined from the Machining_feature that is applying the Directed_taper. The data associated with a Directed_taper are the following:

— angle;

— direction.

4.2.70.1 angle

The angle specifies the amount of slope from the start of the Directed_taper to the end of the Directed_taper. An angle between zero and 90 degrees or between negative 90 and negative 180 degrees indicates that the profile grows larger along the draft direction. An angle between zero and negative 90 degrees or between 90 and 180 degrees indicates that the profile grows smaller along the draft direction See 4.3.74 for the application assertion.

4.2.70.2 direction

The direction specifies the reference direction from which the Directed_taper angle is measured. The direction specifies a vector that points in the direction to apply the taper. See 4.3.73 for the application assertion.

4.2.71 Direction_element

A Direction_element is a type of Shape_element (see 4.2.219) that is a Shape_aspect definition for a direction.

4.2.72 Distance_along_curve_tolerance

A Distance_along_curve_tolerance is a type of Location_tolerance (see 4.2.123) that is the distance calculated between two elements along a path defined by a third element of geometry.

NOTE Figure 43 illustrates the Distance_along_curve_tolerance.

The data associated with a Distance_along_curve_tolerance are the following:

— path.

4.2.72.1 path

The path specifies the shape that the tolerance applies to. See 4.3.75 for the application assertion.

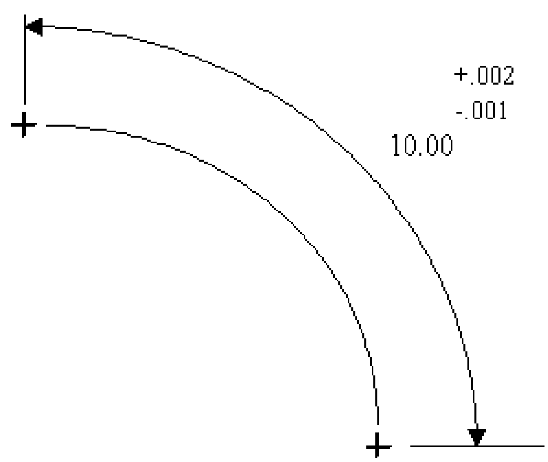


Figure 43 — Distance_along_curve_tolerance

4.2.73 Edge_round

An Edge_round is a type of Transition_feature (see 4.2.258) that is a convex circular arc transition between two intersecting surfaces. The blend surface is tangent to both of the adjacent surface edges. An Edge_round may be a Constant_radius_edge_round (see 4.2.44).

NOTE Figure 44 illustrates the Edge_round with an edge_round_face that is not constant radius.

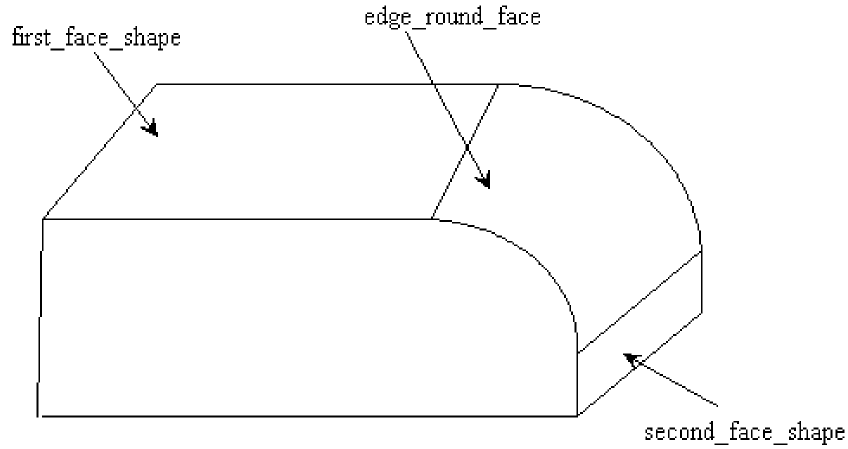


Figure 44 — Edge_round

The data associated with an Edge_round are the following:

— edge_round_face;

— first_face_shape;

— second_face_shape.

4.2.73.1 edge_round_face

The edge_round_face specifies the circular transition surface between the two edges of two surfaces. See 4.3.76 for the application assertion.

4.2.73.2 first_face_shape

The first_face shape specifies one of two surfaces the Edge_round feature will transition between. See 4.3.76 for the application assertion.

4.2.73.3 second_face_shape

The second_face shape specifies the other surfaces the Edge_round feature will transition between. See 4.3.76 for the application assertion.

4.2.74 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_order objects apply only to changes that affect process planning.

The data associated with an Engineering_change_order are the following:

— change_order_number;

— new_version.

4.2.74.1 change_order_number

The change_order_number specifies a unique identification of the Engineering_change_order.

4.2.74.2 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 4.3.77 for the application assertion.

4.2.75 Engineering_change_proposal

An Engineering_change_proposal is a document that describes potential modifications to a part. The data associated with an Engineering_change_proposal are the following:

— change_proposal_number;

— incorporated_proposal.

4.2.75.1 change_proposal_number

The change_proposal_number specifies a unique identification of the Engineering_change_proposal.

4.2.75.2 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 4.3.78 for the application assertion.

4.2.76 Explicit_base_shape_representation

An Explicit_base_shape_representation is a type of Base_shape (see 4.2.7) that is the geometric representation needed to define the shape of the bounded geometry for the part.

NOTE The placement of the Explicit_base_shape_representation is definied by the B-rep geometry.

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:

— B-rep_form;

— explicit_shape;

— user_defined_description.

4.2.76.1 B-rep_form

The B-rep_form specifies the shape that is the representation of the bounded geometry for the part. See 4.3.79 for the application assertion.

4.2.76.2 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 endcut_shape_type shall be one of the following:

— casting;

— composite shape;

— forging;

— user_defined.

NOTE See 4.2.76.2.1 - 4.2.76.2.4 for the definition of each allowable value for endcut_shape_type.

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

4.2.76.2.2 composite_shape:
the base shape of the part is defined with non-metallic materials.

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

4.2.76.2.4 user_defined:
the base shape of the part has non standard description defined by the user.

4.2.76.3 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.

4.2.77 Externally_defined_size_dimension

An Externally_defined_size_dimension is a type of Size_tolerance (see 4.2.223) 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.

4.2.77.1 path

The path specifies the path along which the Thickness_dimension is applied or measured. See 4.3.82 for the application assertion.

4.2.77.2 tolerance_class

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

4.2.77.3 tolerance_definition

The tolerance_definition specifies a file being referenced that contains information about the type of tolerance. See 4.3.83 for the application assertion.

4.2.78 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:

— location_placement;

— identified_by.

4.2.78.1 location_placement

The location_placement specifies the positioning and orientation of the external product data within the geometric domain of the product data of this part of ISO 10303. See 4.3.80 for the application assertion.

4.2.78.2 identified_by

The identified_by specifies the part being referenced, that is contained in an eternal library. See 4.3.81 for the application assertion.

4.2.79 Face_shape_element

A Face_shape_element is a type of Shape_element (see 4.2.219) that is a Shape_aspect definition for a Face.

4.2.80 Face_shape_element_relationship

A Face_shape_element_relationship is the sequence in which face_shape_element objects are applied. The Face_shape_element_relationship defines which face is the preceding face and which face is the succeeding face.

The data associated with a Face_shape_element_relationship are the following:

— predecessor;

— successor.

4.2.80.1 predecessor

The predecessor specifies the Face_shape_element (see 4.2.79) with the highest precedence. See 4.3.84 for the application assertion.

4.2.80.2 successor

The successor specifies the Face_shape_element (see 4.2.79) with the lesser precedence. See 4.3.84 for the application assertion.

4.2.81 Fillet

A Fillet is a type of Transition_feature (see 4.2.258) that is a concave circular arc transition between two intersecting surfaces. The blend surface may be tangent to both of the adjacent surface edges. A Fillet may be a Constant_radius_fillet (see 4.2.45).

NOTE Figure 45 illustrates a Fillet with a fillet face that is not constant radius.

The data associated with a Fillet are the following:

— fillet_face;

— first_face_shape;

— second_face_shape.

4.2.81.1 fillet_face

The fillet_face specifies the circular transition surface between the two edges of two surfaces. See 4.3.85 for the application assertion.

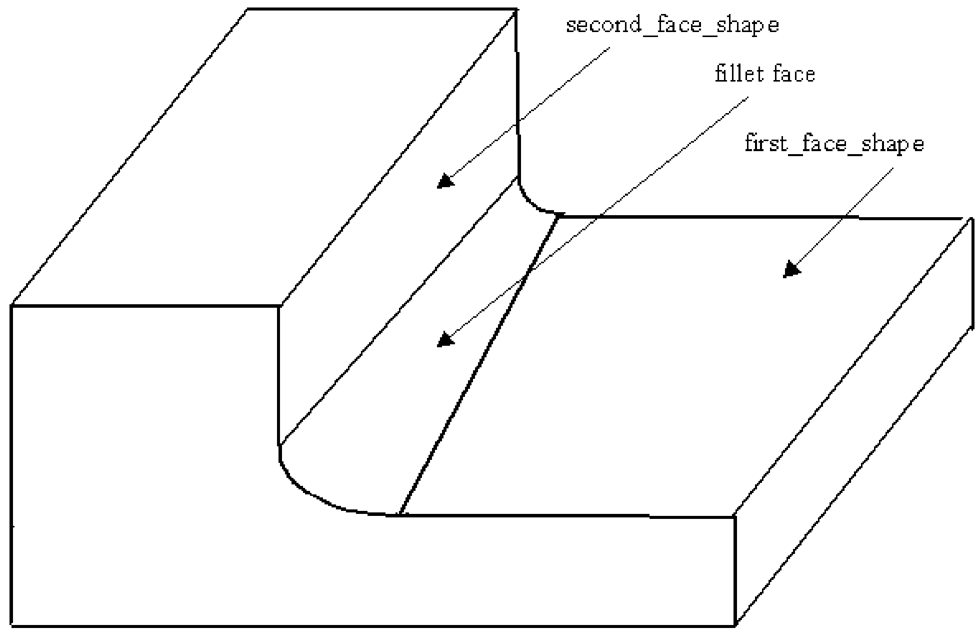


Figure 45 — Fillet

4.2.81.2 first_face_shape

The first_face_shape specifies one of two surfaces the Fillet feature will transition between. See 4.3.85 for the application assertion.

4.2.81.3 second_face_shape

The second_face_shape specifies the second of two surfaces the Fillet feature will transition between. See 4.3.85 for the application assertion.

4.2.82 First_offset

A First_offset is the amount of length offset from a face for creating a Chamfer feature.

NOTE Figure 15 illustrates a First_offset for a Chamfer feature.

The data associated with a First_offset are the following:

— face_shape;

— offset_amount.

4.2.82.1 face_shape

The first_face specifies a geometric shape for one of two faces the Chamfer feature will transition between. See 4.3.86 for the application assertion.

4.2.82.2 offset_amount

The offset_amount specifies a distance from the edge of a face to the start of the Chamfer. See 4.3.87 for the application assertion.

4.2.83 Flat_hole_bottom

A Flat_hole_bottom is a type of Blind_bottom_condition (see 4.2.9) that is the bottom of a Round_hole that shall be flat and have no corner radius.

NOTE Figure 46 illustrates a Flat_hole_bottom for a Round_hole.

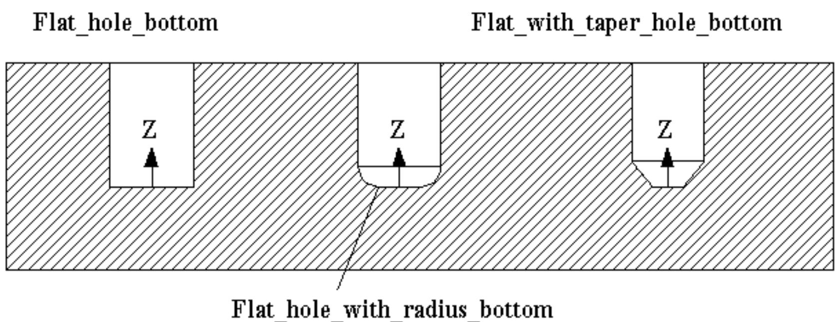


Figure 46 — Flat_hole_bottom

4.2.84 Flat_slot_end_type

A Flat_slot_end_type is a type of Slot_end_type (see 4.2.225) that is an end condition of a slot that shall be a planar shape perpendicular to both of the adjacent Slot wall surfaces. The intersection of the Slot wall surfaces and the end planar shape need not be blended by a radius.

NOTE Figure 47 illustrates a Flat_slot_end_type for a Slot.

The data associated with a Flat_slot_end_type are the following:

— first_radius;

— second_radius.

4.2.84.1 first_radius

The first radius specifies the circular arc transition between the wall of the Slot feature and the planar surface of the Flat_slot_end_type. The position of the first radius is where the Slot wall intersects the Flat_slot_end_type at a positive value along the X-axis. See 4.3.88 for the application assertion.

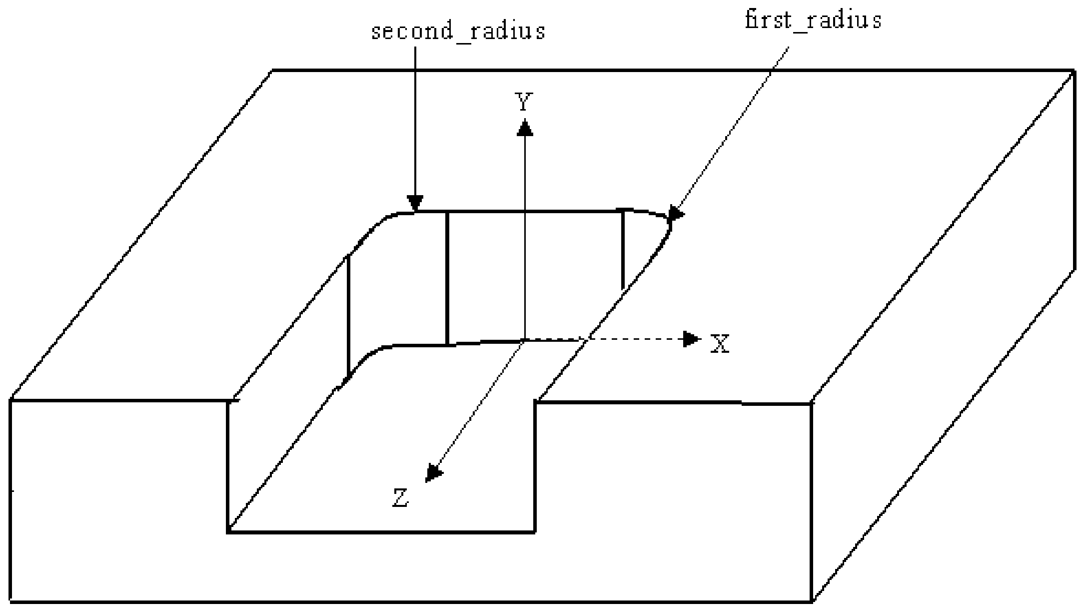


Figure 47 — Flat_slot_end_type

4.2.84.2 second_radius

The second radius specifies the circular arc transition between the wall of the Slot feature and the planar surface of the Flat_slot_end_type. The position of the second radius is where the Slot wall intersects the Flat_slot_end_type at a negative value along the X-axis. See 4.3.88 for the application assertion.

4.2.85 Flat_with_radius_hole_bottom

A Flat_with_radius_hole_bottom is a type of Blind_bottom_condition (see 4.2.9) that the bottom of a Round_hole is flat and has corner radius that are smaller than the diameter of the Round_hole.

NOTE Figure 46 illustrates a Flat_with_radius_hole_bottom for a Round_hole.

The data associated with a Flat_with_radius_hole_bottom are the following:

— corner_radius.

4.2.85.1 corner_radius

The corner_radius specifies the radius between the side and the floor of a Round_hole (see 4.2.212). See 4.3.89 for the application assertion.

4.2.86 Flat_with_taper_hole_bottom

A Flat_with_taper_hole_bottom is a type of Blind_bottom_condition (see 4.2.9) that the bottom of a Round_hole is flat and has a planar taper from the sides of the Round_hole.

NOTE Figure 46 illustrates a Flat_with_taper_hole_bottom for a Round_hole.

The data associated with a Flat_with_taper_hole_bottom are the following:

— final_diameter;

— taper_diameter.

4.2.86.1 final_diameter

The final_diameter specifies the diameter of the Round_hole floor which is a diameter smaller than the initial diameter. See 4.3.90 for the application assertion.

4.2.86.2 taper_diameter

The taper_diameter specifies the angle between the side and the floor of a Round_hole (see 4.2.212), measured along the axis inside the Round_hole. See 4.3.90 for the application assertion.

4.2.87 Flatness_tolerance

A Flatness_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a tolerance for how much a surface is allowed to deviate from being flat. All points of the actual surface shall lie between two parallel planes that are a distance apart equal to the specified tolerance.

NOTE 1 Figure 48 illustrates a Flatness_tolerance.

NOTE 2 The Flatness_tolerance definition is derived from ISO 1101:2004, 18.2.

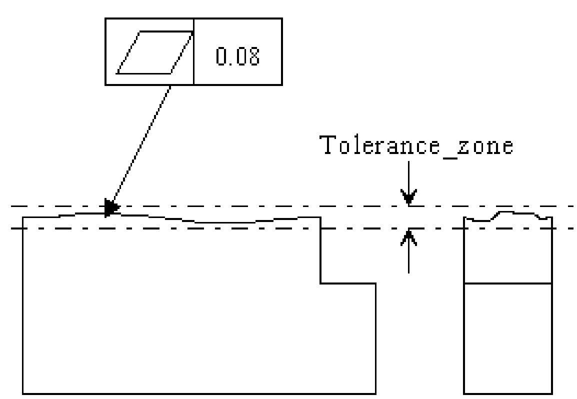


Figure 48 — Flatness_tolerance

The data associated with a Flatness_tolerance are the following:

— segment_size.

4.2.87.1 segment_size

The segment_size specifies the length of a surface to apply a tolerance if the Flatness_tolerance is not applied to the total length. The segment_size need not be specified for a particular Flatness_tolerance.

4.2.88 Gear

A Gear is a type of Machining_feature (see 4.2.126) that is a toothed machine part, such as a wheel or cylinder, that meshes with another toothed part to transmit motion or to change speed or direction. Each Gear is either a Defined_gear (see 4.2.59) or a Catalogue_gear (see 4.2.18).

NOTE vocabulary for gears is defined in ISO 1122-1 and defines the requirements for Gear. Information used to define the data associated with a gear is defined by ISO 1328-1, ISO 1340, and ISO 2203.

The data associated with a Gear are the following:

— applied_shape;

— face_width;

— internal_or_external_gear;

— module_or_diameter_pitch;

— nominal_tooth_depth;

— normal_attribute;

— number_of_teeth;

— profile_shift;

— reference_pressure_angle;

— root_fillet_radius;

— tip_diameter.

4.2.88.1 applied_shape

The applied_shape specifies a base shape for applying the Gear feature. See for 4.3.91 the application assertion.

4.2.88.2 face_width

The face_width specifies a width over the toothed part of a gear, measured along a generator of the reference cylinder. See 4.3.92 for the application assertion.

NOTE Figure 49 illustrates a face_width for a Gear.

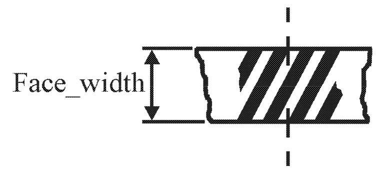


Figure 49 — Gear face_width

4.2.88.3 internal_or_external_gear

The internal_or_external_gear specifies whether or not the gear teeth are applied to an internal surface or an external surface.

4.2.88.4 module_or_diameter_pitch

The module_or_diameter_pitch specifies which of either the module or diametral pitch of a gear is specified by the normal_attribute (see 4.2.88.6).

The module of a spur gear is the quotient of the pitch at the reference surface divided by π. The normal module of a helical gear is the quotient of the normal pitch divided by π. The normal pitch is the pitch at the reference surface measured along the arc of a co-cylindrical normal helix, lying between the tooth traces of consecutive corresponding flanks.

The diametral pitch is the quotient of the number π divided by the pitch, or the quotient of the number of teeth divided by the reference diameter. The normal diametral pitch of a helical gear is the quotient of the number π divided by the normal pitch, or the quotient of the number of teeth divided by the product of the reference diameter and the cosine of the helix angle.

4.2.88.5 nominal_tooth_depth

The nominal_tooth_depth specifies the radial distance between the tooth tip and root circles. See 4.3.92 for the application assertion.

NOTE Figure 50 illustrates the nominal_tooth_depth for a Gear.

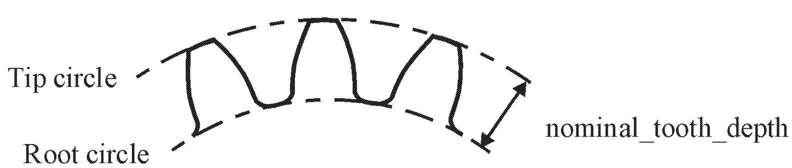


Figure 50 — Gear nominal_tooth_depth

4.2.88.6 normal_attribute

The normal_attribute specifies the module when the value of module_or_diameter_pitch is 'module' or the diametral pitch when the value of module_or_diameter_pitch is 'diameter pitch'. For a spur gear, the normal_attribute is the module or diametral pitch, requiring no qualification; for a helical gear, the normal_attribute specifies the normal module or the normal diametral pitch. See 4.3.92 for the application assertion.

4.2.88.7 number_of_teeth

The number_of_teeth specifies the number of teeth in the circumference produced on the part surface. See 4.3.92 for the application assertion.

4.2.88.8 profile_shift

The profile_shift is the quotient of the rack shift divided by the module (see 4.2.88.4). The rack shift specifies the distance measured along a common normal between the reference cylinder of the gear and the datum plane of the basic rack, when the rack and the gear are superposed so that the flanks of a tooth of one are tangent to those of the other. By convention, the profile shift is positive when the datum plane is external to the cylinder and negative when it cuts it. This definition is valid for both external and internal gears. For internal gears, tooth profiles are considered to be those of the tooth spaces. See 4.3.92 for the application assertion.

NOTE Figure 51 illustrates a profile_shift for a Gear.

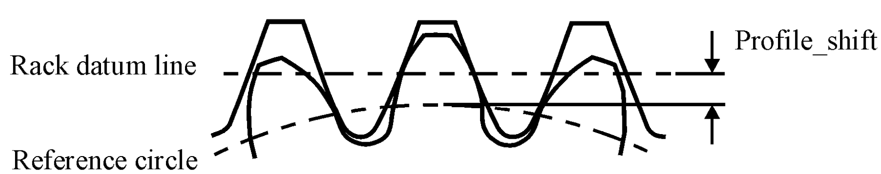


Figure 51 — Gear profile_shift

4.2.88.9 reference_pressure_angle

The reference_pressure_angle specifies the acute angle between a radial line passing through the point of intersection of the profile with the reference circle and a tangent to the profile at that point. See 4.3.92 for the application assertion.

NOTE Figure 52 illustrates a reference_pressure_angle for a Gear.

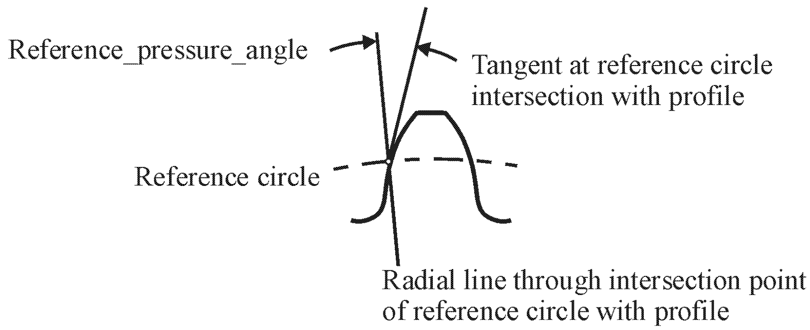


Figure 52 — Gear reference_pressure_angle

4.2.88.10 root_fillet_radius

The root_fillet_radius specifies the acceptable radius of the surface between the usable flank and the root surface. See 4.3.92 for the application assertion.

NOTE Figure 53 illustrates a root_fillet_radius for a Gear.

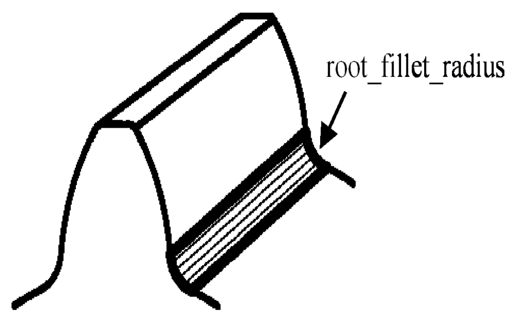


Figure 53 — Gear root_fillet_radius

4.2.88.11 tip_diameter

The tip_diameter specifies the diameter of the tip circle. See 4.3.92 for the application assertion.

NOTE Figure 54 illustrates a tip_diameter for a Gear.

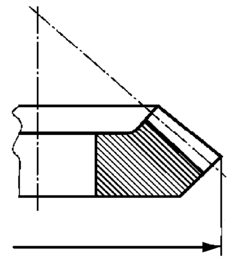


Figure 55

Figure 54 — Gear tip_diameter

4.2.89 General_boss

A General_boss is a type of Boss (see 4.2.11) that is an enclosed volume bounded by an arbitrary shape.

NOTE Figure 55 illustrates the General_boss.

The data associated with a General_boss are the following:

— change_in_boundary;

— enclosed_boundary.

4.2.89.1 change_in_boundary

The change_in_boundary specifies a taper that defines the change in shape of the General_boss. The change_in_boundary need not be specified for a particular General_boss. See 4.3.94 for the application assertion.

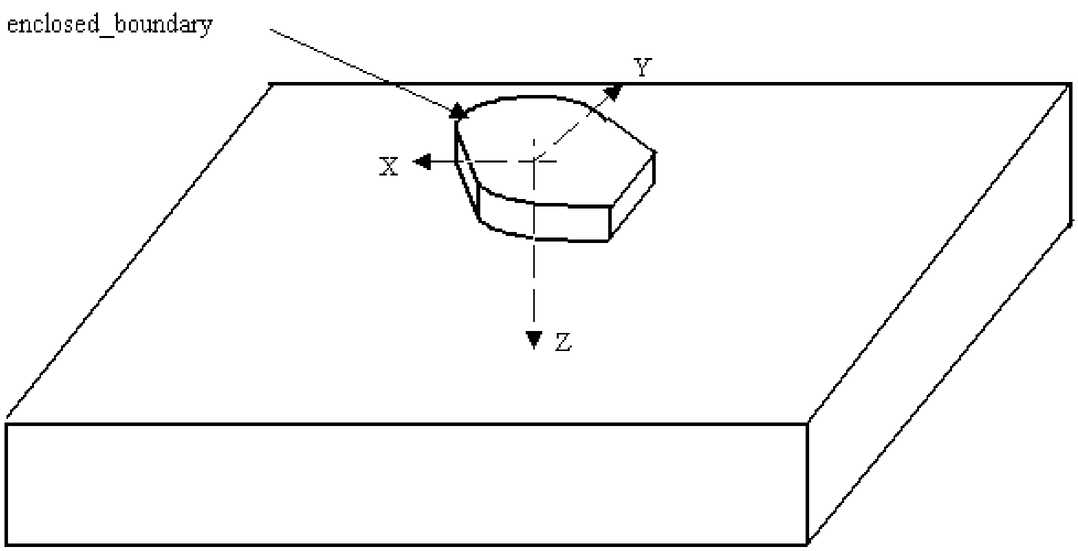


Figure 55 — General_boss

4.2.89.2 enclosed_boundary

The enclosed_boundary specifies an outline or shape that bounds an enclosed area with no opening for a General_boss. The Closed_profile specifies the area required by a General_boss. The placement of the enclosed_boundary shall be with the origin of the Closed_profile at the origin of the General_boss. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_boss. See 4.3.93 for the application assertion.

4.2.90 General_closed_profile

A General_closed_profile is a type of Closed_profile (see 4.2.35) that is an enclosed area bounded by a arbitrary shape. The orientation is defined by the explicit geometry of the shape.

NOTE Figure 56 illustrates a General_closed_profile.

The data associated with a General_closed_profile are the following:

— closed_profile_shape.

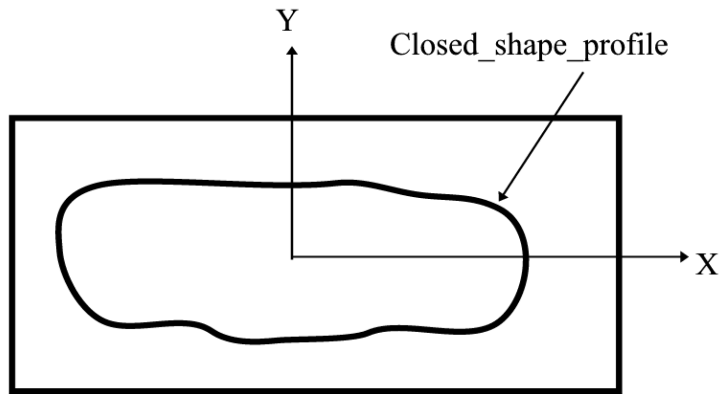


Figure 56 — General_closed_profile

4.2.90.1 closed_profile_shape

The closed_profile_shape specifies a closed curve that defines the arbitrary shape of the profile. See 4.3.96 for the application assertion.

4.2.91 General_cutout

A General_cutout is a type of Cutout (see 4.2.50) that is a volume of arbitrary shape removed from the part and shall pass through two faces of the part. A General_cutout is similar in definition to a General_-pocket, but differ in the type of process required to manufacture. The data associated with a Circular_-cutout are the following:

— boundary.

4.2.91.1 boundary

The boundary specifies an outline or shape that is an area that may be closed or partially open. The profile specifies the area required by a General_cutout. The placement of the boundary shall be with the origin of the Profile at the origin of the General_cutout. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_cutout. See 4.3.97 for the application assertion.

4.2.92 General_open_profile

A General_open_profile is a type of Open_profile (see 4.2.145) that is specified by a shape bounded by an arbitrary planar shape. The orientation is defined by the explicit geometry of the shape.

NOTE Figure 57 illustrates a General_open_profile.

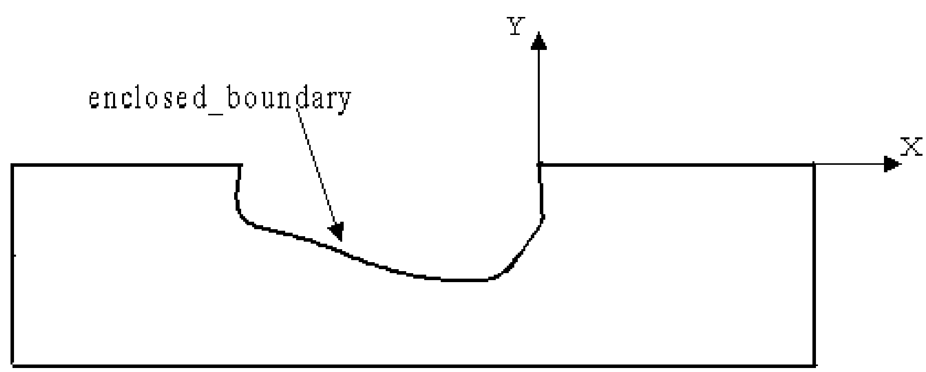


Figure 57 — General_open_profile

The data associated with a General_open_profile are the following:

— enclosed_boundary.

4.2.92.1 enclosed_boundary

The enclosed_boundary specifies a curve with no enclosing bounds that defines the arbitrary planar shape of the profile. See 4.3.98 for the application assertion.

4.2.93 General_outside_profile

A General_outside_profile is a type of Profile_feature (see 4.2.182) that is specified the removal volume of raw stock or other excess material of arbitrary shape from the outside shape of the part. The General_outside_profile feature may remove material from the entire outside shape or some portion of the shape. A single part may have several General_outside_profile objects, with the Z-axes of the coordinate systems of the General_outside_profile objects pointing in any direction.

NOTE Figure 58 illustrates a General_outside_profile.

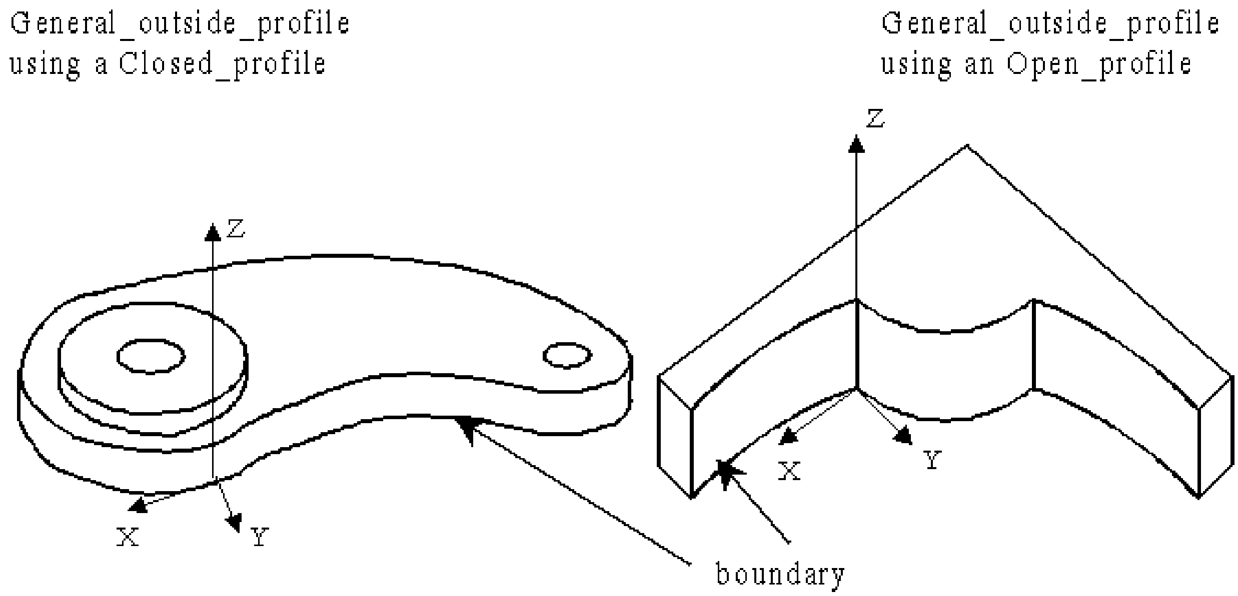


Figure 58 — General_outside_profile

The data associated with a General_outside_profile are the following:

— boundary.

4.2.93.1 boundary

The boundary specifies a profile that identifies the outside shape of the part. The placement of the boundary shall be with the origin of the profile at the origin of the General_outside_profile feature. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the feature. See 4.3.99 for the application assertion.

4.2.94 General_path

A General_path is a type of Path (see 4.2.163) that is a direction of travel along an arbitrary curve.

NOTE Figure 59 illustrates a Slot feature with a Square_u_profile and a General_path. The path has the same orientation as the Slot feature. The orientation of the profile positions it to the orientation of the Slot feature. The Square_U_profile has orientation to position it at the same position as the Slot feature.

The data associated with a General_path are the following:

— sweep_path.

4.2.94.1 sweep_path

The sweep_path specifies a continues set of curves that define an arbitrary direction of travel. See 4.3.100 for the application assertion.

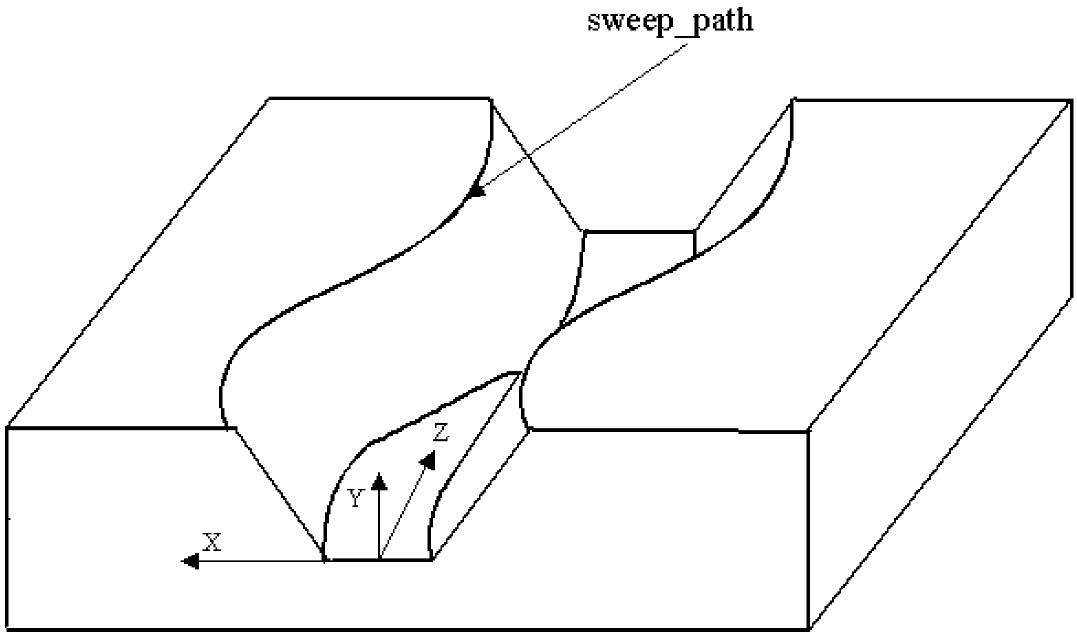


Figure 59 — General_path

4.2.95 General_pattern

A General_pattern is a type of Replicate_feature (see 4.2.204) that is a base shape component and a list of arbitrary placements to arrange identical copies of the base feature. The placement of all instances of the base feature are relative to the Replicate_feature coordinate system.

NOTE Figure 60 illustrates a General_pattern of Round_hole features.

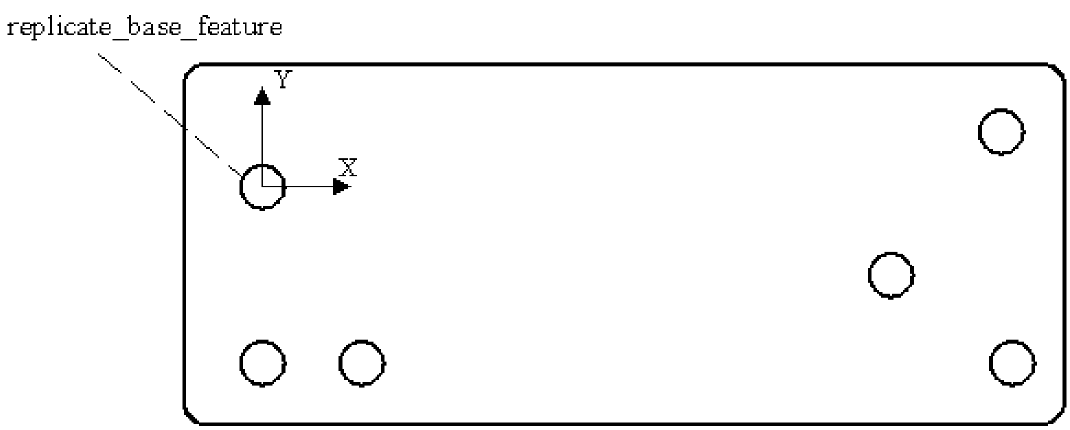


Figure 60 — General_pattern

The data associated with a General_pattern are the following:

— feature_placement.

4.2.95.1 feature_placement

The feature_placement specifies a set of axis and positions to place a base feature in the General_pattern. A General_pattern is defined as one base feature and many placements allowing the base feature to be placed several times in a random pattern. There may be more than one feature_placement for a General_pattern. See 4.3.101 for the application assertion.

4.2.96 General_pocket

A General_pocket is a type of Pocket (see 4.2.177) that is a volume of arbitrary shape removed from the part.

NOTE Figure 61 illustrates a General_pocket with a Through_pocket_bottom_condition and a pocket depth that defines the highest point of the pocket.

The data associated with a General_pocket are the following:

— boundary;

— volume_not_removed.

4.2.96.1 boundary

The boundary specifies an outline or shape that is an area that may be closed or partially open. The profile specifies the area required by a General_pocket. The placement of the boundary shall be with the origin of the Profile at the origin of the General_pocket. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_pocket. See 4.3.102 for the application assertion.

4.2.96.2 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss feature defines the shape of the material that is to remain in the pocket. See 4.3.103 or 4.3.104 for the application assertion.

4.2.97 General_pocket_bottom_condition

A General_pocket_bottom_condition is a type of Pocket_bottom_condition (see 4.2.178) that specifies an enclosed area bounded by an arbitrary shape.

NOTE Figure 61 illustrates the General_pocket_bottom_condition applied to a General_pocket with an Open_profile.

The data associated with a General_pocket_bottom_condition are the following:

— floor;

— floor_radius.

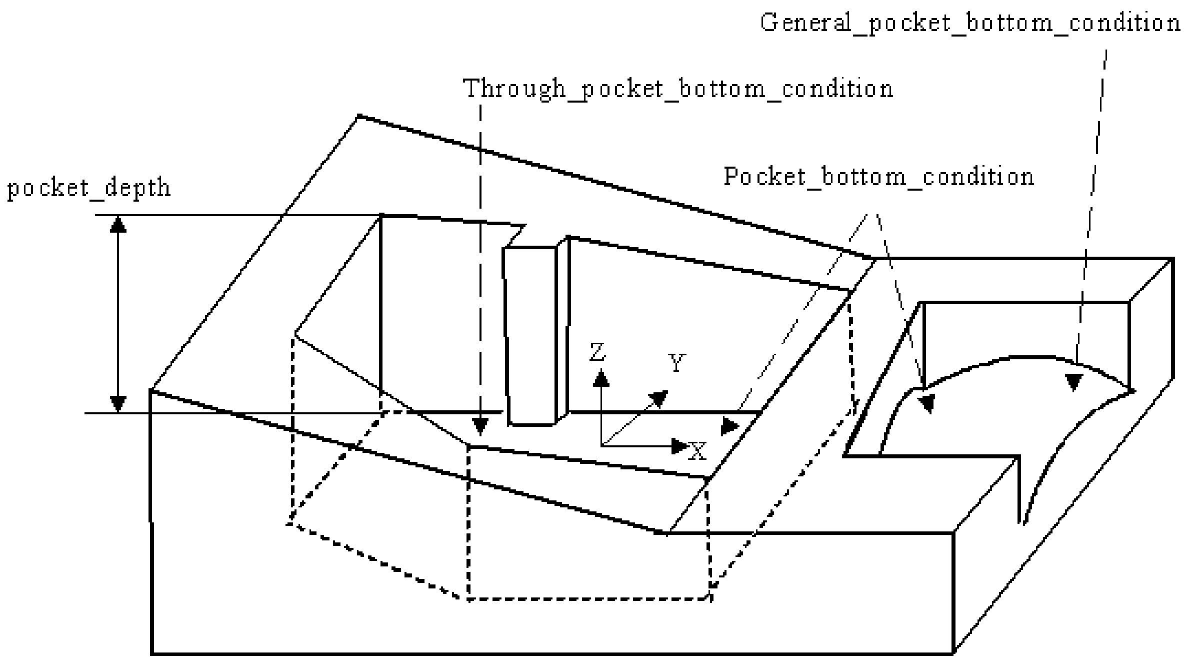


Figure 61 — General_pocket

4.2.97.1 floor

The floor specifies the face at the bottom of a Pocket feature, adjacent to all the pocket walls. See 4.3.105 for the application assertion.

4.2.97.2 floor_radius

The floor radius is specifies the amount of curvature for an arc between the bottom and the sides of a pocket feature. See 4.3.106 for the application assertion.

4.2.98 General_profile_floor

A General_profile_floor is a type of Profile_floor (see 4.2.183) that specifies an enclosed area bounded by an arbitrary shape.

NOTE Figure 107 illustrates a Shape_profile with a General_profile_floor.

The data associated with a General_profile_floor are the following:

— floor.

4.2.98.1 floor

The floor specifies the face at the bottom of a General_profile_floor feature, adjacent to all the Shape_profile walls. See 4.3.107 for the application assertion.

4.2.99 General_removal_volume

A General_removal_volume is a type of Multi_axis_feature(see 4.2.140) that is an enclosed volume of arbitrary shape that shall be removed from the part. The position and orientation shall be determined from the shape defining geometry.

NOTE Figure 62 illustrates a General_removal_volume.

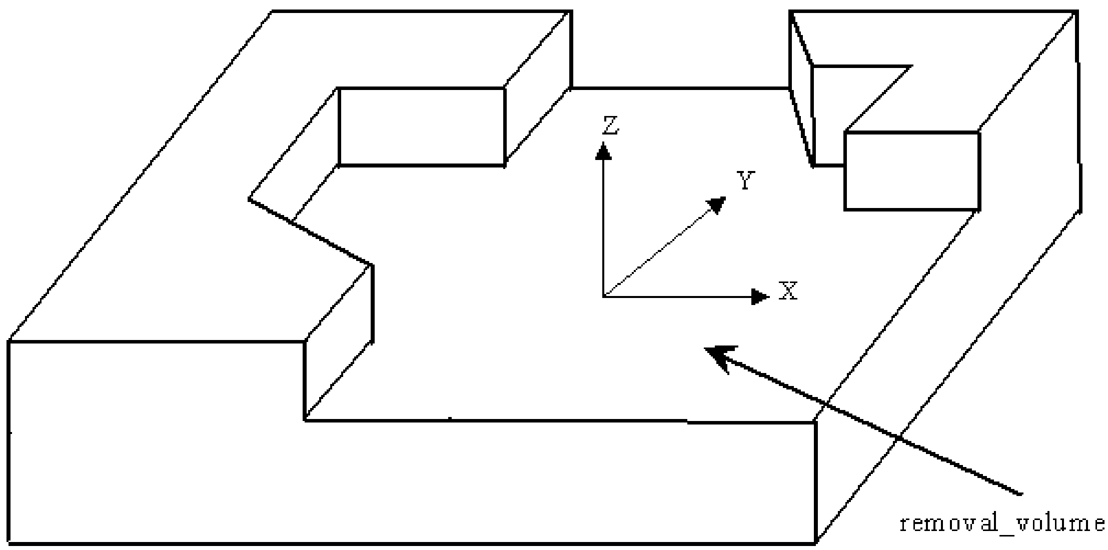


Figure 62 — General_removal_volume

The data associated with a General_removal_volume are the following:

— removal_volume.

4.2.99.1 removal_volume

The removal_volume specifies the arbitrary shape to be removed. See 4.3.108 for the application assertion.

4.2.100 General_revolution

A General_revolution is a type of Revolved_feature (see 4.2.207) that is an arbitrary planar shape swept one complete revolution about an axis. The arbitrary planar shape shall be finite in length, coplanar with the axis of revolution, and shall not intersect the axis of revolution. The General_revolution may be either an outer shape of a part or a volume removal, depending on the material direction.

NOTE Figure 63 illustrates a General_revolution for an outer shape, the material direction points to the material side. Figure 64 illustrates a General_revolution for a volume removal.

The data associated with a General_revolution are the following:

— outer_edge_shape.

4.2.100.1 outer_edge_shape

The outer_edge_shape specifies an outline or shape that shall be revolved about an axis. The General_open_profile specifies the outer edge shape required by a General_revolution. The placement of the outer_edge_shape shall be on the Y-axis of the General_revolution at a specified distance away from the origin of the General_revolution. The X-axis and Y-axis of the profile shall be the same as the Y-axis and Z-axis of the General_revolution. See 4.3.109 for the application assertion.

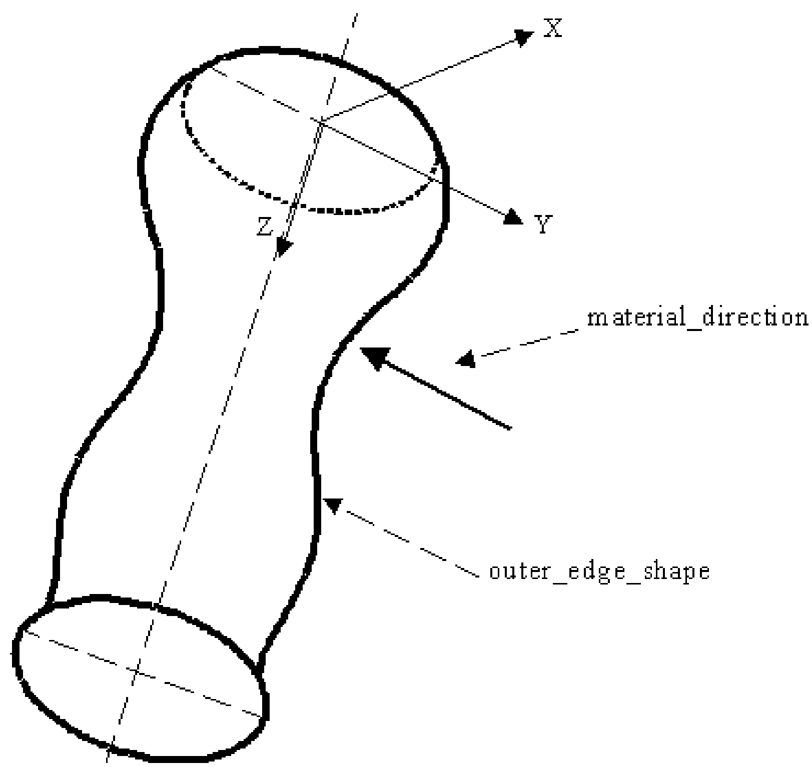


Figure 63 — General_revolution for an outer shape

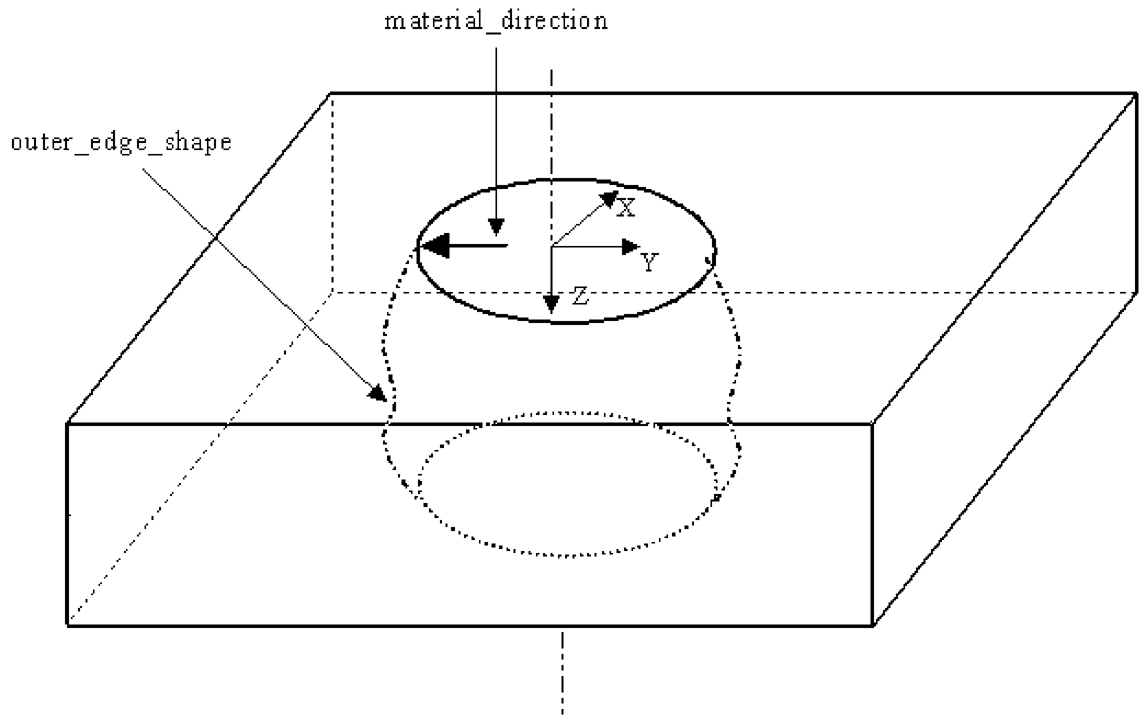


Figure 64 — General_revolution for a volume removal

4.2.101 General_rib_top_floor

A General_rib_top_floor is a type of Rib_top_floor (see 4.2.211) that specifies an enclosed area bounded by an arbitrary shape. The floor defining shapes may be defined in a particular sequence. The data associated with a General_rib_top_floor are the following:

— rib_top_face.

4.2.101.1 rib_top_face

A rib_top_face specifies a set of faces at the bottom of a Rib_top (see 4.2.210). The order of the faces is achieved using a Feature_shape_element_relationship (see 4.2.80). See 4.3.110 for the application assertion.

4.2.102 General_shape_profile

A General_shape_profile is a type of Shape_profile (see 4.2.220) that is a volume of arbitrary shape which defines a portion of the part. The data associated with a General_shape_profile are the following:

— profile_boundary.

4.2.102.1 profile_boundary

The profile_boundary specifies the outline of the Shape_profile feature. The outline defines an area that may or may not be entirely enclosed. The placement of the profile_boundary shall be with the origin of the Path, that defines the profile, at the origin of the General_shape_profile. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the General_shape_profile. See 4.3.111 for the application assertions.

4.2.103 General_top_condition

A General_top_condition is a type of Boss_top_condition (see 4.2.12) that specifies an enclosed area bounded by an arbitrary shape that defines the top of a Boss feature. The data associated with a General_-top_condition are the following:

— top_face.

4.2.103.1 top_face

The top_face specifies a face at the top of a Boss feature, adjacent to the Boss sides. See 4.3.112 for the application assertion.

4.2.104 Geometric_tolerance

A Geometric_tolerance is the maximum or minimum variation from true geometric form or position that may be permitted in manufacture. Geometric_tolerance should be employed only for those requirements of a part critical to its functioning or interchangability. Each Geometric_tolerance is one of the following: an Angularity_tolerance (see 4.2.5), a Circular_runout_tolerance (see 4.2.32),a Circularity_tolerance (see 4.2.33), a Concentricity_tolerance (see 4.2.42), a Cylindricity_tolerance (see 4.2.53), a Flatness_tolerance (see 4.2.87), a Linear_profile_tolerance (see 4.2.120), a Parallelism_tolerance (see 4.2.154), a Perpendicularity_tolerance (see 4.2.166), a Position_tolerance (see 4.2.179), a Straightness_tolerance (see 4.2.235), a Surface_profile_tolerance (see 4.2.236), a Symmetry_tolerance (see 4.2.239), or a Total_-runout_tolerance (see 4.2.257).

NOTE Geometric tolerance definitions are derived from ISO 1101 and ANSI Y14.5[2].

The data associated with a Geometric_tolerance are the following:

— applied_shape;

— geometric_tolerance_value;

— modifier_control;

— significant_digits;

— unit_of_measure;

— zone_definition.

4.2.104.1 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 4.3.114 for the application assertion.

4.2.104.2 geometric_tolerance_value

The geometric_tolerance_value specifies the tolerance amount that a part is allowed to meet the required accuracy for proper fit.

4.2.104.3 modifier_control

The modifier_control specifies the material condition which is applied to the shape being toleranced by the Geometric_tolerance. The modifier_control need not be specified for a particular Geometric_-tolerance. See 4.3.113 for the application assertion.

4.2.104.4 significant_digits

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

4.2.104.5 unit_of_measure

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

4.2.104.6 zone_definition

The zone_definition specifies the tolerance zone that restricts the Geometric_tolerance. The zone_-definition need not be specified for a particular Geometric_tolerance. See 4.3.115 for the application assertion.

4.2.105 Geometric_tolerance_precedence_relationship

A Geometric_tolerance_precedence_relationship is a composite geometric tolerance. A part may have a shape that has a geometric tolerance and that shape is the basic shape for a pattern which may also have a geometric tolerance. The geometric tolerance for the base shape shall have precedence over the geometric tolerance for the pattern. The data associated with a Geometric_tolerance_precedence_-relationship are the following:

— base_shape_tolerance;

— pattern_shape_tolerance.

4.2.105.1 base_shape_tolerance

The base_shape_tolerance specifies the Geometric_tolerance which is applied to a shape on a Part. See 4.3.116 for the application assertion.

4.2.105.2 pattern_shape_tolerance

The pattern_shape_tolerance specifies the Geometric_tolerance which is applied to a pattern on a Part. See 4.3.116 for the application assertion.

4.2.106 Groove

A Groove is a type of Revolved_feature (see 4.2.207) that is a narrow channel or depression that is swept through one complete revolution about an axis.

NOTE Figure 65 illustrates two Groove features. The face shape that has the Groove applied to it is determined by the profile orientation.

The data associated with a Groove are the following:

— sweep.

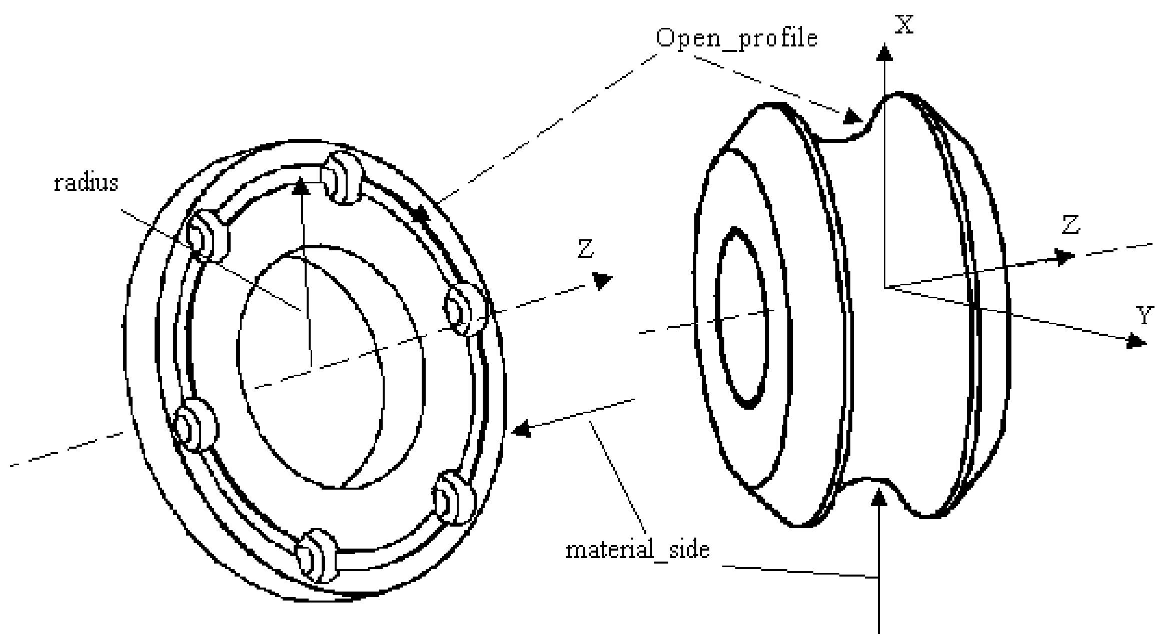


Figure 65 — Groove

4.2.106.1 sweep

The sweep specifies an outline or shape that shall be revolved about an axis. The Open_profile specifies the sweep shape required by a Groove. The placement of the profile shall be along the X-axis of the Groove at a specified distance away from the origin. The orientation of the Open_profile is independent of the orientation of the Groove feature. The Groove feature may be defined on different faces of a part depending on the orientation of the profile. See 4.3.117 for the application assertion.

4.2.107 Hardness

A Hardness is the resistance of a material surface to deformation by external forces. The data associated with a Hardness are the following:

— high_value;

— low_value;

— nominal;

— scale.

4.2.107.1 high_value

The high_value specifies the highest allowed value of hardness for a specific material type.

4.2.107.2 low_value

The low_value specifies the lowest allowed value of hardness for a specific material type.

4.2.107.3 nominal

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

4.2.107.4 scale

The scale specifies the method of determining hardness.

EXAMPLE Rockwell and Brinell are examples of scale.

4.2.108 Height_dimension

A Height_dimension is a type of Size_tolerance (see 4.2.223) that specifies the size along a straight line that is referred to as height in the referenced shape. The data associated with a Height_dimension are the following:

— path.

4.2.108.1 path

The path specifies the path along which the Height_dimension is applied or measured. See 4.3.118 for the application assertion.

4.2.109 Helical_bevel_gear

The Helical_bevel_gear is a type of Bevel_gear(see 4.2.8) where the tooth traces are non-cylindrical helices, joining the definitions of a s of a Bevel_gear and a Helical_gear.

The data associated with an Helical_bevel_gear are the following:

— left_or_right_tooth;

— reference_helix_angle.

4.2.109.1 left_or_right_tooth

The left_or_right_hand_tooth specifies a description of whether the gear teeth are right or left handed.

Right hand teeth are teeth whose successive transverse profiles show clockwise displacement with increasing distance from an observer looking along the straight line generators of the reference surface

Left hand teeth are teeth whose successive transverse profiles show anti-clockwise displacement with increasing distance from an observer looking along the straight line generators of the reference surface.

4.2.109.2 reference_helix_angle

The reference_helix_angle is the acute angle between the tangent to the tooth trace of a straight helical gear and the straight generator of the reference cylinder on which it lies. See 4.3.119 for the application assertion.

4.2.110 Helical_gear

A Helical_gear is a type of Defined_gear (see 4.2.59) that is a cylindrical gear whose tooth traces are helices.

NOTE - Figure 66 illustrates the Helical_gear.

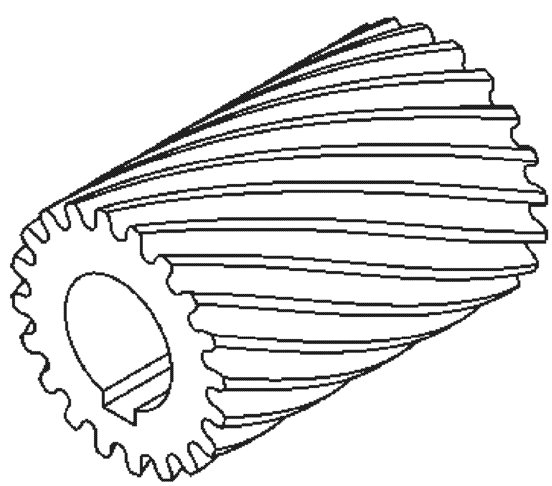


Figure 66 — Helical_gear

The data associated with an Helical_gear are the following:

— left_or_right_tooth;

— reference_helix_angle.

4.2.110.1 left_or_right_tooth

The left_or_right_hand_tooth specifies a description of whether the gear teeth are right or left handed.

Right hand teeth are teeth whose successive transverse profiles show clockwise displacement with increasing distance from an observer looking along the straight line generators of the reference surface

Left hand teeth are teeth whose successive transverse profiles show anti-clockwise displacement with increasing distance from an observer looking along the straight line generators of the reference surface.

NOTE Figure 67 illustrates left or right hand teeth for a Helical_gear.

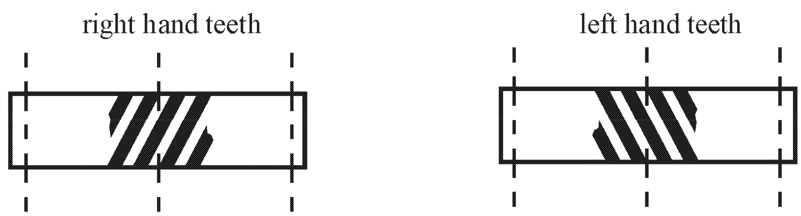


Figure 67 — Helical_gear left_or_right_hand_tooth

4.2.110.2 reference_helix_angle

The reference_helix_angle is the acute angle between the tangent to the tooth trace of a straight helical gear and the straight generator of the reference cylinder on which it lies. See 4.3.120 for the application assertion.

NOTE Figure 68 illustrates a reference_helix_angle for a Helical_gear.

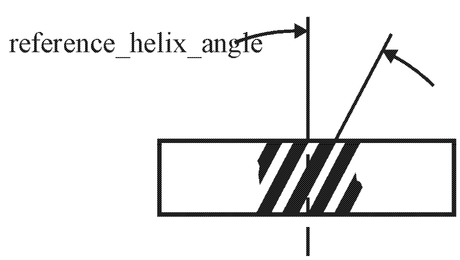


Figure 68 — Helical_gear reference_helix_angle

4.2.111 Hole

A Hole is a type of Multi_axis_feature(see 4.2.140) that is the removal of a cylindrical volume from a part. Each Hole is either a Counterbore_hole (see 4.2.46), a Countersunk_hole (see 4.2.47), or a Round_-hole (see 4.2.212). The Hole may be positioned at its bottom with the Z-axis in the direction out of the Hole or at a position at the top of the hole with the Z-axis in the direction into the part.

4.2.112 Implicit_base_shape_representation

An Implicit_base_shape_representation is a type of Base_shape (see 4.2.7) that is the type of representation needed to define the shape type and minimum size required for 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 4.2.10), a Cylindrical_base_shape (see 4.2.52), or a Ngon_base_shape (see 4.2.141).

NOTE Figure 69 illustrates types of Implicit_base_shape_representations.

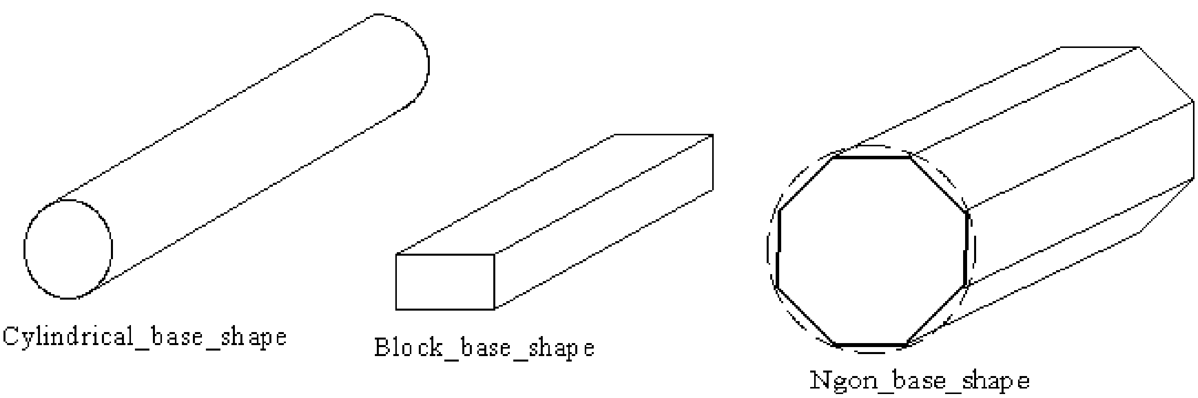


Figure 69 — Implicit_base_shape_representation

The data associated with an Implicit_base_shape_representation are the following:

— base_shape_length;

— placement.

4.2.112.1 base_shape_length

The base_shape_length is the size of the length of a Implicit_base_shape_representation. See 4.3.121 for the application assertion.

4.2.112.2 placement

A placement specifies the positioning of the part part orientation within the Implicit_base_shape_-representation. The positioning will be different for different types of Implicit_base_shape_-representation. See 4.3.122 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.

4.2.113 Indirect_stock_requisition

An Indirect_stock_requisition is a type of Requisition (see 4.2.205) that defines indirect items required to support part manufacturing.

4.2.114 Knurl

A Knurl is a type of Machining_feature (see 4.2.126) that is a scoring pattern made by a series of small ridges or beads on a metal surface. Each Knurl is either a Catalogue_knurl (see 4.2.19) or a Turned_knurl (see 4.2.259).

NOTE Figure 70 illustrates a Knurl types.

The data associated with a Knurl are the following:

— applied_shape;

— partial_profile.

4.2.114.1 applied_shape

The applied_shape specifies a base shape for applying the Knurl feature. See for 4.3.124 the application assertion.

4.2.114.2 partial_profile

The partial_profile specifies the placement and length of a surface to apply a Knurl feature. The partial_profile need not be specified for a particular Knurl. See 4.3.123 for the application assertion.

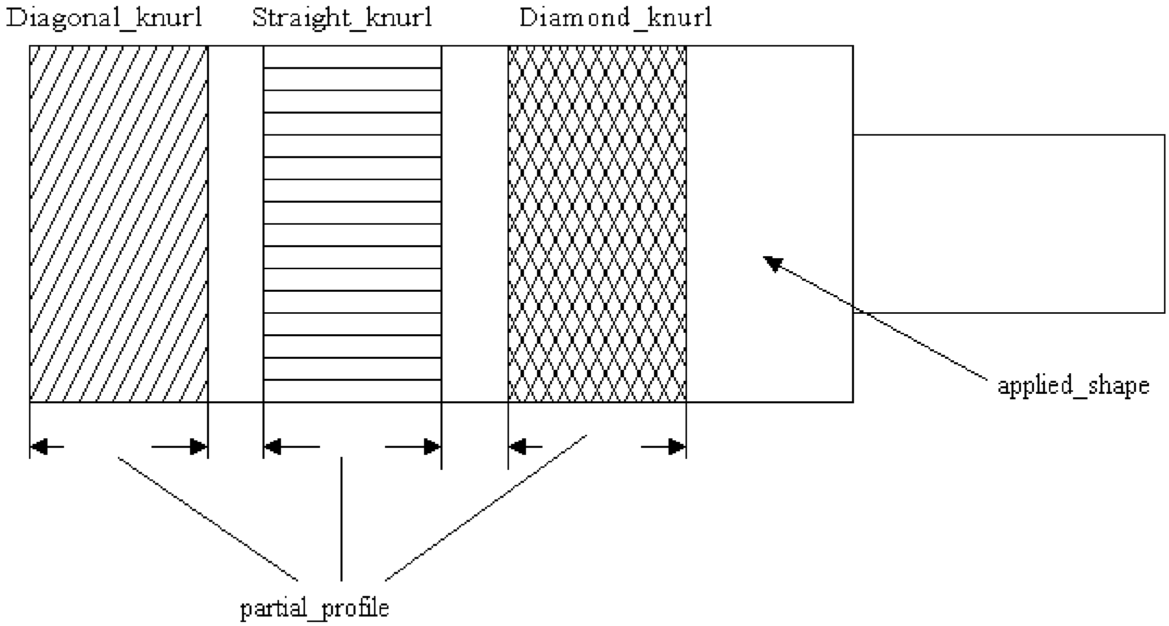


Figure 70 — Knurl

4.2.115 Length_dimension

A Length_dimension is a type of Size_tolerance (see 4.2.223) that specifies the size along a straight line that is referred to as length in the referenced shape. The data associated with a Length_dimension are the following:

— path.

4.2.115.1 path

The path specifies the path along which the Length_dimension is applied or measured. See 4.3.125 for the application assertion.

4.2.116 Library_part_assignment

An Library_part_assignment provides the means to reference information about a class within a parts library dictionary. The data associated with an Library_part_assignment are the following:

— definitional_class_BSU;

— definitional_property_value_pairs.

4.2.116.1 definitional_class_BSU

The definitional_class_BSU specifies the identification of the component within a parts library as defined by ISO-13584. See 4.3.127 for the application assertion.

4.2.116.2 definitional_property_value_pairs

The definitional_property_value_pairs specifies the set of pairs (Property_BSU (see 4.2.187), Property_value(see 4.2.189)) defining the properties of the class. See 4.3.126 for the application assertion.

4.2.117 Limits_and_fits

A Limits_and_fits contains the necessary information to express a tolerance of the limits-and-fits system standardized by ISO 286. The data associated with a Limits_and_fits are the following:

— deviation;

— fitting_type;

— grade.

NOTE A limits and fits system is definied in ISO 286-1 and defines the requirements for Limits and fits.

4.2.117.1 deviation

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

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

4.2.117.2 fitting_type

The fitting_type specifies whether the tolerance declaration applies to a shaft or to a hole. The fitting type need not be specified for a particular Limits_and_fits.

4.2.117.3 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 which is definied in ISO 286-2 and defines the requirements for grade.

4.2.118 Linear_path

A Linear_path is a type of Path (see 4.2.163) that is a direction of travel along a line.

NOTE Figure 71 illustrates a Machining feature with a Linear_path. This path has the same orientation as the feature.

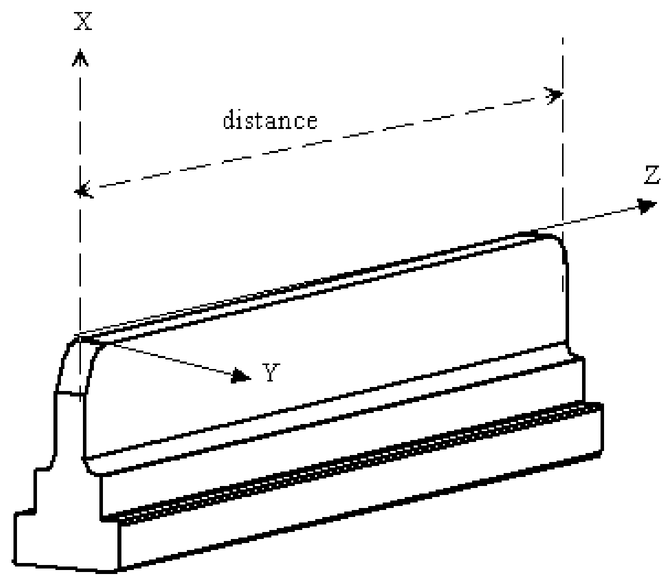


Figure 71 — Linear_path

The data associated with a Linear_path are the following:

— direction;

— distance.

4.2.118.1 direction

The direction specifies a vector which indicates the the direction of the path starting from the path placement. See 4.3.129 for the application assertion.

4.2.118.2 distance

The distance specifies the length of the path. See 4.3.128 for the application assertion.

4.2.119 Linear_profile

A Linear_profile is a type of Open_profile (see 4.2.145) that is a straight line of a specified length. The Linear_profile shall have orientation parallel to the X-axis.

NOTE Figure 72 illustrates a Linear_profile that is being applied to the Planar_face feature.

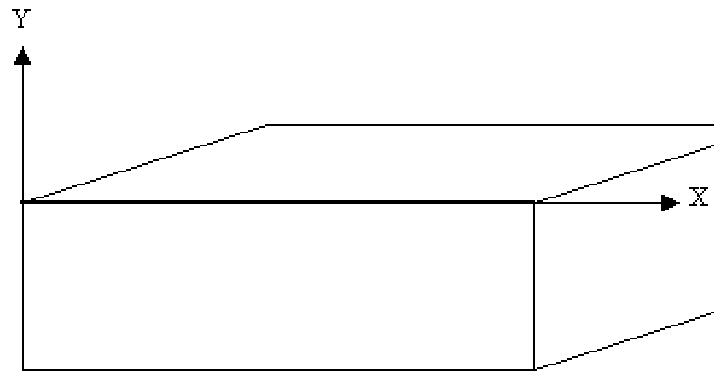


Figure 72 — Linear_profile

The data associated with a Linear_profile are the following:

— profile_length.

4.2.119.1 profile_length

The profile_length specifies the length of the profile. See 4.3.130 for the application assertion.

4.2.120 Linear_profile_tolerance

A Linear_profile_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTE 1 Figure 73 illustrates the Linear_profile_tolerance

NOTE 2 The Linear_profile_tolerance definition is derived from ISO 1101:2004, 18.5.

The data associated with a Linear_profile_tolerance are the following:

— affected_plane;

— geometric_reference.

4.2.120.1 affected_plane

The affected_plane specifies the plane to apply the Linear_profile_tolerance. The affected_plane need not be specified for a particular Linear_profile_tolerance. See 4.3.132 for the application assertion.

4.2.120.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.131 for the application assertion.

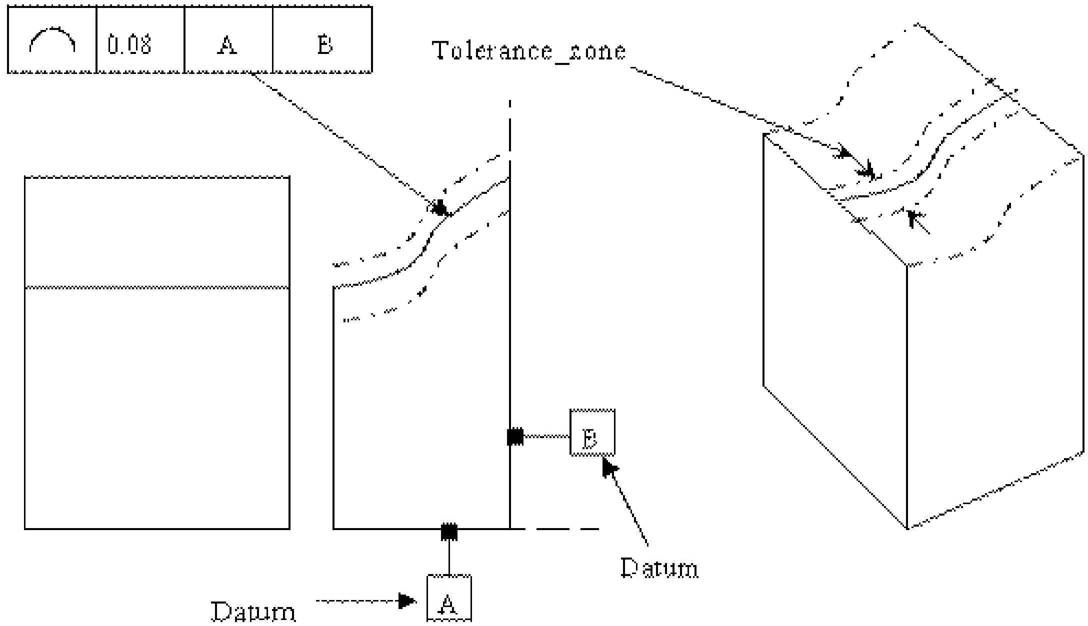


Figure 73 — Linear_profile_tolerance

4.2.121 Location_dimension_tolerance

A Location_dimension_tolerance is a type of Location_tolerance (see 4.2.123) that is the allowable variation in locating one feature of a part with respect to another.

NOTE Figure 74 illustrates a Location_dimension_tolerance.

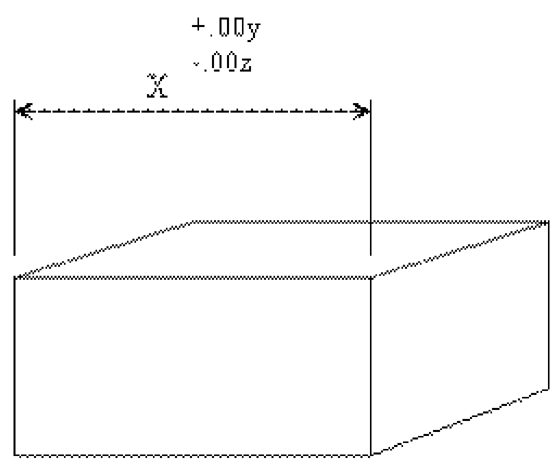


Figure 74 — Location_dimension_tolerance

The data associated with a Location_dimension_tolerance are the following:

— plane_and_direction.

4.2.121.1 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 4.3.133 for the application assertion.

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

4.2.122 Location_element

A Location_element is a type of Shape_element (see 4.2.219) that is a Shape_aspect representation for a reference point.

4.2.123 Location_tolerance

A Location_tolerance is a type of Dimensional_tolerance (see 4.2.69) 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 4.2.3), a Location_dimension_tolerance (see 4.2.121), or a Distance_along_curve_tolerance (see 4.2.72). The data associated with a Location_-tolerance are the following:

— directed;

— origin_shape;

— termination_shape.

4.2.123.1 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.

4.2.123.2 origin_shape

The origin_shape specifies the shape on the Part that defines the starting position for a Location _tolerance. See 4.3.134 for the application assertion.

4.2.123.3 termination_shape

The termination_shape specifies the shape on the Part that defines the ending position for a Location_tolerance. See 4.3.134 for the application assertion.

4.2.124 Loop_slot_end_type

A Loop_slot_end_type is the end conditions of a slot that is a closed loop. The start point and end point of the Slot are the same point, the Slot has no end openings.

4.2.125 Machine_requisition

A Machine_requisition is a type of Requisition (see 4.2.205) that describes machines required to support part manufacturing.

4.2.126 Machining_feature

A Machining_feature is a type of Manufacturing_feature (see 4.2.129) that identifies a volume of material that shall be removed to obtain the final part geometry from the initial stock. Machining_features requires both direction and location in placing them on a part. Each Machining_feature may be one of the following: a Knurl (see 4.2.114), a Multi_axis_feature(see 4.2.140), an Outer_round (see 4.2.153), a Revolved_feature (see 4.2.207), a Thread (see 4.2.246), a Marking (see 4.2.131), a Spherical_cap (see 4.2.228), or a Compound_feature (see 4.2.39). The data associated with a Machining_feature are the following:

— placement;

— usage_name.

4.2.126.1 placement

The placement specifies the position and orientation of a Machining_feature relative to the base shape for a part. See 4.3.135 for the application assertion.

4.2.126.2 usage_name

The usage_name specifies a user defined name that is additional information about the use of a feature. The usage_name need not be specified for a particular Machining_feature.

4.2.127 Manufactured_assembly

A Manufactured_assembly is a type of Part (see 4.2.155) 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 Figure 75 illustrates the Manufactured_assembly and Mating_definition

4.2.128 Manufactured_assembly_relationship

A Manufactured_assembly_relationship identifies the Manufactured_assembly, and a component of the assembly. A component is either Single_piece_part or another Manufactured_assembly.

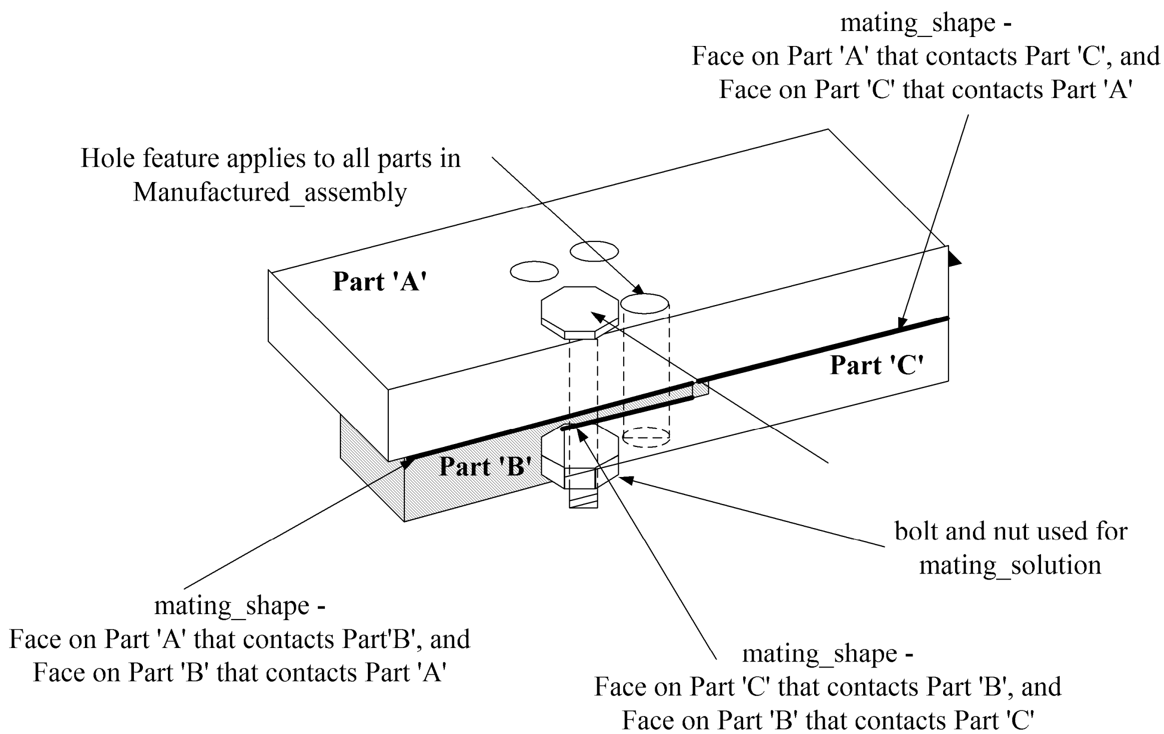


Figure 75 — Manufactured_assembly and Mating_definition

The data associated with a Manufactured_assembly_relationship are the following:

— assembly;

— component;

— orientation.

4.2.128.1 assembly

The assembly specifies the Manufactured_assembly that shall have a component of Single_piece_part or another Manufactured_assembly. See 4.3.136 for the application assertion.

4.2.128.2 component

The component specifies either a Single_piece_part or another Manufactured_assembly used to define an assembly. See 4.3.137 for the application assertion.

4.2.128.3 orientation

The orientation specifies the transformation of a Part to define its placement in a manufacturing assembly, or sub-assembly. See 4.3.138 for the application assertion.

4.2.129 Manufacturing_feature

A Manufacturing_feature is a type of Shape_element (see 4.2.219) that identifies the types of features necessary to manufacture a machined part. Each Manufacturing_feature is either a Machining_feature (see 4.2.126), a Replicate_feature (see 4.2.204), or a Transition_feature (see 4.2.258).

4.2.130 Manufacturing_feature_group

A Manufacturing_feature_group specifies the collection of manufacturing features with a usage identification. A Manufacturing_feature_group shall allow for the collection of collections.

EXAMPLE A Manufacturing_feature_group may be used to group together all of the hold down holes for a part.

The data associated with a Manufacturing_feature_group are the following:

— feature_groups;

— group_description;

— group_name.

4.2.130.1 feature_groups

The feature_group specifies the list of Manufacturing_feature (see 4.2.129) or the Manufacturing_-feature_group (see 4.2.130) to be grouped. See 4.3.139 and 4.3.140 for the application assertion.

4.2.130.2 group_description

The group_description specifies additional text information about the Manufacturing_feature_group.

4.2.130.3 group_name

The group_name specifies identification for the Manufacturing_feature_group.

4.2.131 Marking

A Marking is a type of Machining_feature (see 4.2.126) that is one or more text characters on a surface of a part. Each Marking is either a Defined_marking (see 4.2.60) or a Catalogue_marking (see 4.2.20).

The data associated with a Marking are the following:

— applied_to_shape;

— text.

4.2.131.1 applied_to_shape

The applied_to_shape specifies a base shape for applying the Marking feature. See 4.3.142 for the application assertion.

4.2.131.2 text

The text specifies the characters that will be applied to the part. See 4.3.141 for the application assertion.

4.2.132 Material

A Material is the identification of the raw stock from which a part is produced. Material identifies primary and substitution material. The data associated with a Material are the following:

— material_characteristics;

— material_description;

— material_id;

— material_specification;

— stock_size.

4.2.132.1 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 4.3.143 for the application assertion.

4.2.132.2 material_description

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

4.2.132.3 material_id

The material_id specifies a word or group of words that make up the unique designation of the material.

4.2.132.4 material_specification

The material_specification specifies the documentation which contains additional information about Material. The material_specification need not be specified for a particular Material. There may be more than one material_specification for a Material. See 4.3.144 for the application assertion.

4.2.132.5 stock_size

The stock_size specifies the dimensions of the raw material required to make the part. The stock_size need not be specified for a particular Material.

4.2.133 Material_condition_modifier

A Material_condition_modifier is a description of a condition for a part surface, it is an indication whether the maximum material principle is to be applied respectively to the toleranced feature, the datum feature, or both. Tolerances of position are determined by a material condition of a mating surface. The data associated with a Material_condition_modifier are the following:

— material_type.

4.2.133.1 material_type

The material_type specifies the type of condition allowed for the Material_condition_modifier.

EXAMPLES Maximum material condition (MMC), Least material condition (LMC) and Regardless of feature size (RFS).

4.2.134 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;

— property_characteristic.

4.2.134.1 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 4.3.145 for the application assertion.

4.2.134.2 property_characteristic

The property_characteristic specifies the parameter for the description of Material_property. The property_characteristics need not be specified for a particular Material_property. There may be more than one property_characteristics for a Material_property. See 4.3.146 for the application assertion.

4.2.135 Material_requisition

A Material_requisition is a type of Requisition (see 4.2.205) which describes an order for purchase of raw materials by the manufacturing organization.

4.2.136 Mating_definition

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

NOTE Figure 75 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;

— standard_part_fastener.

4.2.136.1 applied_assembly

An applied_assembly specifies the Manufactured_assembly that contains the Single_piece_parts that have a mating definition. See 4.3.148 for the application assertion.

4.2.136.2 mating_shape

The mating_shape specifies the shape that form the area of mating contact between two single_piece_part objects. See 4.3.149 for the application assertion.

4.2.136.3 mating_solution

The mating_solution specifies additional Single_piece_parts that participate in the Mating_definition. See 4.3.150 for the application assertion.

EXAMPLE Two parts can 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.

4.2.136.4 mating_type

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

4.2.136.5 standard_part_fastener

A standard_part_fastener specifies a libraries that define representation categories capable of being provided for any family of parts. See 4.3.147 for the application assertion.

NOTE The libraries are external files, in ISO 13584 format, which are referenced by part of ISO 10303.

EXAMPLE Examples of standard_part_fasteners are washers, bolds and nuts.

4.2.137 Mating_definition_relationship

A Mating_definition_relationship specifies additional information about the mating of two particular Single_piece_parts 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.

4.2.137.1 mated_part

The mated_part specifies the single_piece_part that shall have a mating definition with another single_piece_part. See 4.3.153 for the application assertion.

4.2.137.2 mating_part_definition

The mating_part_definition specifies the mating definition for two single_piece_parts that contact each other in an assembly. See 4.3.151 for the application assertion.

4.2.137.3 orientation

The orientation specifies the transformation of a part to define its placement in a mating definition. A Mating_definition_relationship may but need not require an orientation to be specified. See 4.3.152 for the application assertion.

4.2.138 Mating_relationship

A Mating_relationship defines two single_piece_parts that are in the same Manufactured_assembly and are in contact with each other. The data associated with a Mating_relationship are the following:

— predecessor;

— successor.

4.2.138.1 predecessor

The predecessor specifies the Single_piece_part with the highest precedence. See 4.3.154 for the application assertion.

4.2.138.2 successor

The predecessor specifies the Single_piece_part with the lesser precedence. See 4.3.154 for the application assertion.

4.2.139 Modular_fixture_requisition

A Modular_fixture_requisition is a type of Requisition (see 4.2.205) which describes an order for purchase of modular fixtures by the manufacturing organization.

4.2.140 Multi_axis_feature

A Multi_axis_feature is a type of Machining_feature (see 4.2.126) that identifies milling features for a part, and not turned features. Each Multi_axis_feature may be one of the following: a Boss (see 4.2.11), a General_removal_volume (see 4.2.99), a Hole (see 4.2.111), a Rounded_end (see 4.2.213), a Planar_face (see 4.2.171), a Pocket (see 4.2.177), a Profile_feature (see 4.2.182), a Protrusion (see 4.2.190), a Rib_top (see 4.2.210), a Slot (see 4.2.224), and a Step (see4.2.232). The data associated with Multi_axis_feature are the following:

— maximum_feature_limit.

4.2.140.1 maximum_feature_limit

The maximum_feature_limit specifies a planar limitation for a feature. No portion of the feature shall exist beyond this planar definition. The normal to the plane shall be in the direction away from the Machining_feature. See 4.3.155 for the application assertion.

4.2.141 Ngon_base_shape

An Ngon_base_shape is a type of Implicit_base_shape_representation (see 4.2.112) that specifies the initial shape of the material is a polygon with any number of sides.

NOTE Figure 76 illustrates a Ngon_base_shape.

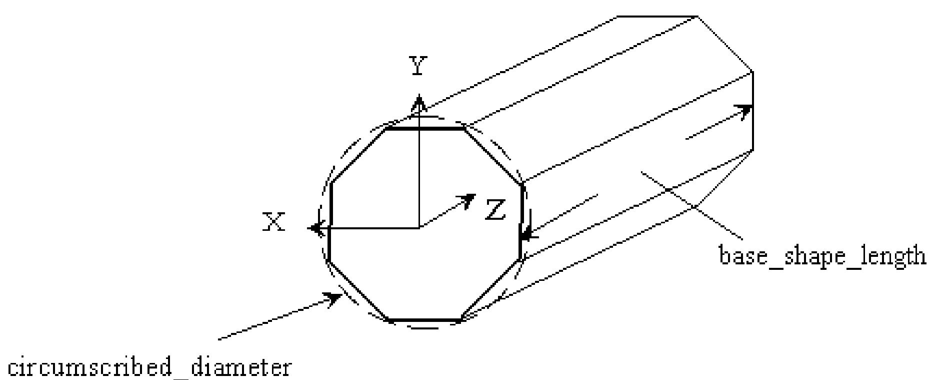


Figure 76 — Ngon_base_shape

The data associated with a Ngon_base_shape are the following:

— circumscribed_or_across_flats;

— corner_radius;

— diameter;

— number_of_sides.

4.2.141.1 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.

4.2.141.2 corner_radius

The corner_radius specifies the size of an arc blend between two sides of the ngon. See 4.3.156 for the application assertion.

4.2.141.3 diameter

The diameter specifies specifies the size of either the circumscribed diameter, or the diameter across the flats. See 4.3.155 for the application assertion.

4.2.141.4 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon. See 4.3.156 for the application assertion.

4.2.142 Ngon_profile

An Ngon_profile is a type of Closed_profile (see 4.2.35) that is an enclosed area bounded by three or more connected straight line sides. The orientation is at the center of the profile with one side of the ngon parallel to the X-axis crossing the Y-axis at a negative value.

NOTE Figure 77 illustrates the Ngon_profile.

The data associated with a Ngon_profile are the following:

— circumscribed_or_across_flats;

— corner_radius;

— diameter;

— number_of_sides.

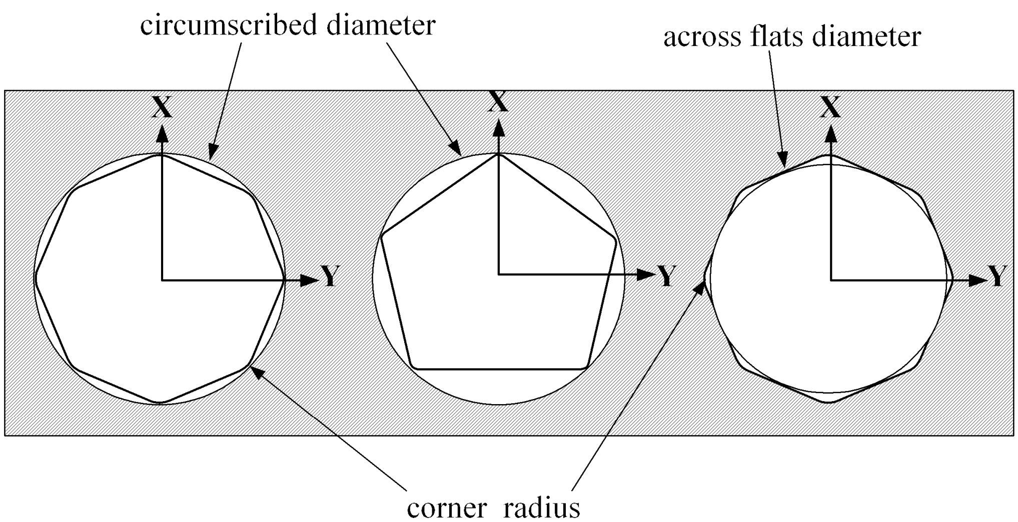


Figure 77 — Ngon_profile

4.2.142.1 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.

4.2.142.2 corner_radius

The corner_radius specifies the size of an arc blend between two sides of the ngon. See 4.3.157 for the application assertion.

4.2.142.3 diameter

The diameter specifies specifies the size of either the circumscribed diameter, or the diameter across the flats. See 4.3.156 for the application assertion.

4.2.142.4 number_of_sides

The number_of_sides specifies how many sides are needed for the Ngon. See 4.3.157 for the application assertion.

4.2.143 Numeric_parameter

A Numeric_parameter is a type of Property_parameter (see 4.2.188) that 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 4.2.144). The data associated with a Numeric_parameter are the following:

— parameter_units;

— parameter_value.

4.2.143.1 parameter_units

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

EXAMPLE watt, meters, degrees, etc.

4.2.143.2 parameter_value

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

4.2.144 Numeric_parameter_with_tolerance

A Numeric_parameter_with_tolerance is a type of Numeric_parameter (see 4.2.143) with an implied tolerance value.

NOTE 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.

4.2.144.1 implicit_tolerance

The implicit_tolerance specifies the type of tolerance to apply to a numeric parameter value. See 4.3.158, 4.3.159, 4.3.160 and 4.3.161 for the application assertion.

4.2.145 Open_profile

An Open_profile is a type of Profile (see 4.2.181) that is an outline or shape with no enclosing or confining bounds. The open ends of the profile may extend infinitely. Each Open_profile is either a General_open_profile (see 4.2.92), a Linear_profile (see 4.2.119), a Partial_circular_profile (see 4.2.161), a Rounded_U_profile (see 4.2.214), a Square_U_profile (see 4.2.231), a Tee_profile (see 4.2.245), or a Vee_profile (see 4.2.260). The data associated with an Open_profile are the following:

— profile_limit.

4.2.145.1 profile_limit

The profile_limit specifies a planar limitation for a feature. No portion of the profile shall exist beyond this planar definition. The normal to the plane shall be in the direction away from the Open_profile. See 4.3.164 for the application assertion.

4.2.146 Open_slot

An Open_slot is a type of Slot (see 4.2.224) that has a course of travel defined by a Path that is open and has openings. The data associated with a Open_slot are the following:

— course_of_travel;

— end_conditions.

4.2.146.1 course_of_travel

The course_of_travel specifies a 3D space curve, that when combine with a Profile, creates the shape of the Slot. See 4.3.162 for the application assertion.

4.2.146.2 end_conditions

The end_conditions specifies the type of implicit shape at the ends of the Slot. See 4.3.163 for the application assertion.

4.2.147 Open_slot_end_type

An Open_slot_end_type is a type of Slot_end_type (see 4.2.225) that is an end condition of a slot that shall pass through the end of the part.

NOTE Figure 78 illustrates a slot with two Open_slot_end_type objects.

4.2.148 Ordered_part

An Ordered_part specifies the number of machined parts, of a particular order number, a Customer_order is requesting. The data associated with an Ordered_part are the following:

— quantity_required;

— quantity_unit_of_measure.

4.2.148.1 quantity_required

The quantity_required specifies the number of items to be manufactured.

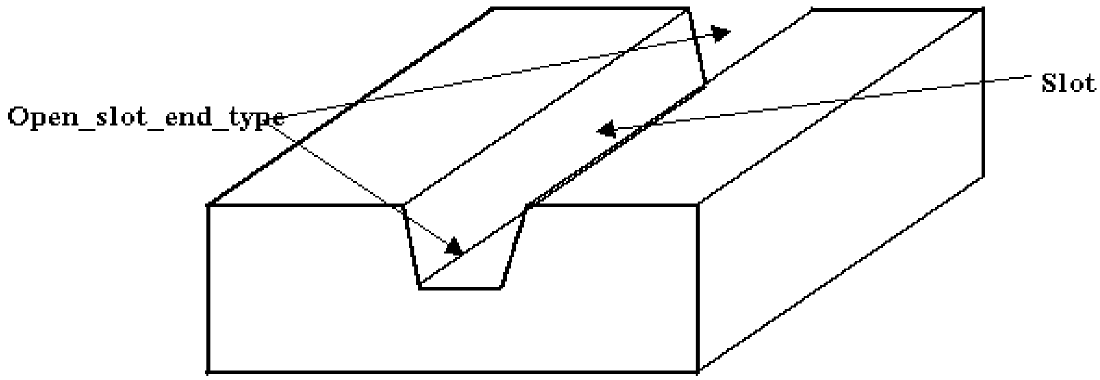


Figure 78 — Open_slot_end_type

4.2.148.2 quantity_unit_of_measure

The quantity_unit_of_measure specifies the units in terms of which the part is expressed.

EXAMPLE Examples of units of measure are: each, meter, watt, etc.

4.2.149 Organization

An Organization is a functional group that is involved in the production of a part. The data associated with an Organization are the following:

— organization_address;

— organization_id;

— organization_name.

4.2.149.1 organization_address

The organization_address specifies a location at which an organization exists and may be contacted. This location may be physical or electronic.

4.2.149.2 organization_id

The organization_id specifies the unique identification of an organization.

4.2.149.3 organization_name

The organization_name specifies a word or group of words by which an organization is commonly referred. This name need not be unique within an organization.

4.2.150 Orientation

An Orientation is the direction and location of the basic shape of a part, feature on the part, or of the component of a feature which are Profile objects and Path objects. The data associated with an Orientation are the following:

— axis;

— location.

4.2.150.1 axis

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

4.2.150.2 location

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

4.2.151 Outer_diameter

An Outer_diameter is a type of Outer_round (see 4.2.153) that is a sweeping of an outline specified by a line segment one complete revolution about an axis. The line is finite in length, coplanar with the axis. An Outer_diameter may have a constant diameter around the axis of rotation, or it may be tapered.

NOTE Figure 79 illustrates the Outer_diameter with and without a taper.

The data associated with an Outer_diameter are the following:

— diameter;

— feature_length;

— reduced_size.

4.2.151.1 diameter

The diameter specifies the maximum diametric size of an Outer_diameter feature. See 4.3.165 for the application assertion.

4.2.151.2 feature_length

The length specifies the size of a Outer_diameter feature, measured along the feature's axis.
See 4.3.164 for the application assertion.

4.2.151.3 reduced_size

The reduced_size specifies the constant change in the Outer_diameter along the feature length. See 4.3.166, 4.3.167 and 4.3.168 for the application assertion.

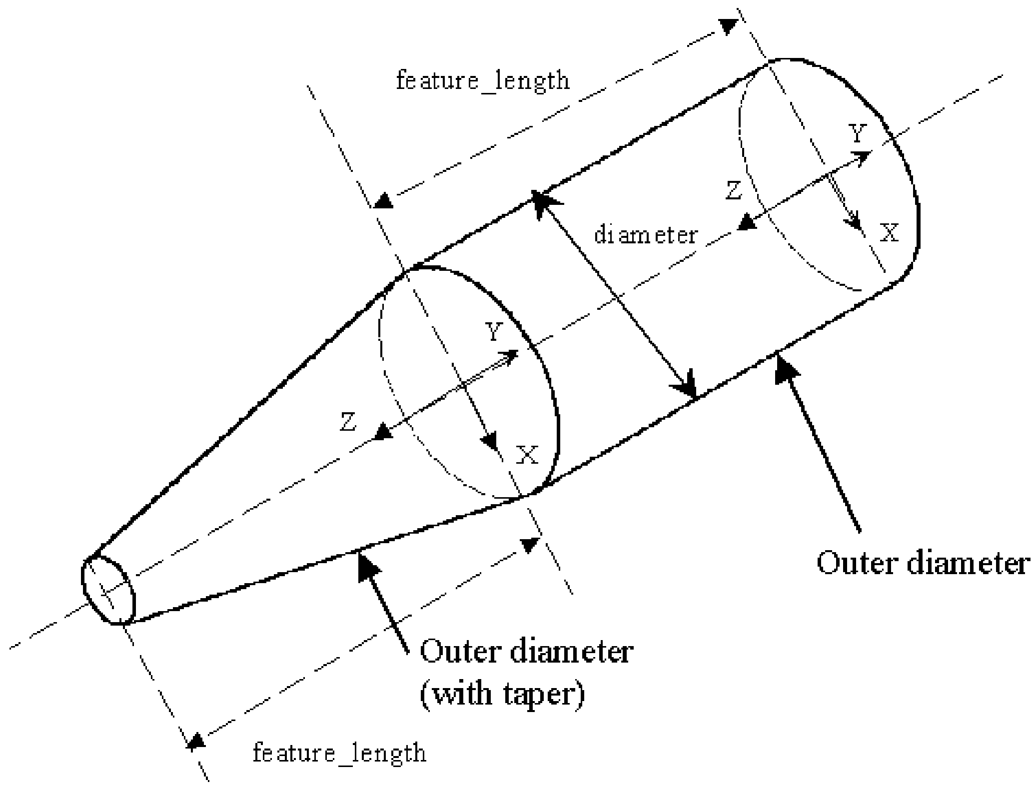


Figure 79 — Outer_diameter

4.2.152 Outer_diameter_to_shoulder

An Outer_diameter_to_shoulder is a type of Outer_round (see 4.2.153) that is a sweeping of a shape one complete revolution about an axis. The shape shall be specified by two lines that connect at a point and extend infinitely. The enclosed angle shall be smaller than a straight angle. The intersection of the two lines need not be blended with a radius.

NOTE Figure 80 illustrates the Outer_diameter_to_shoulder.

The data associated with an Outer_diameter_to_shoulder are the following:

— diameter;

— feature_length;

— v_shape_boundary.

4.2.152.1 diameter

The diameter specifies the size of the part at the point of the Vee, or where the two sides come together, swept about an axis of rotation. See 4.3.169 for the application assertion.

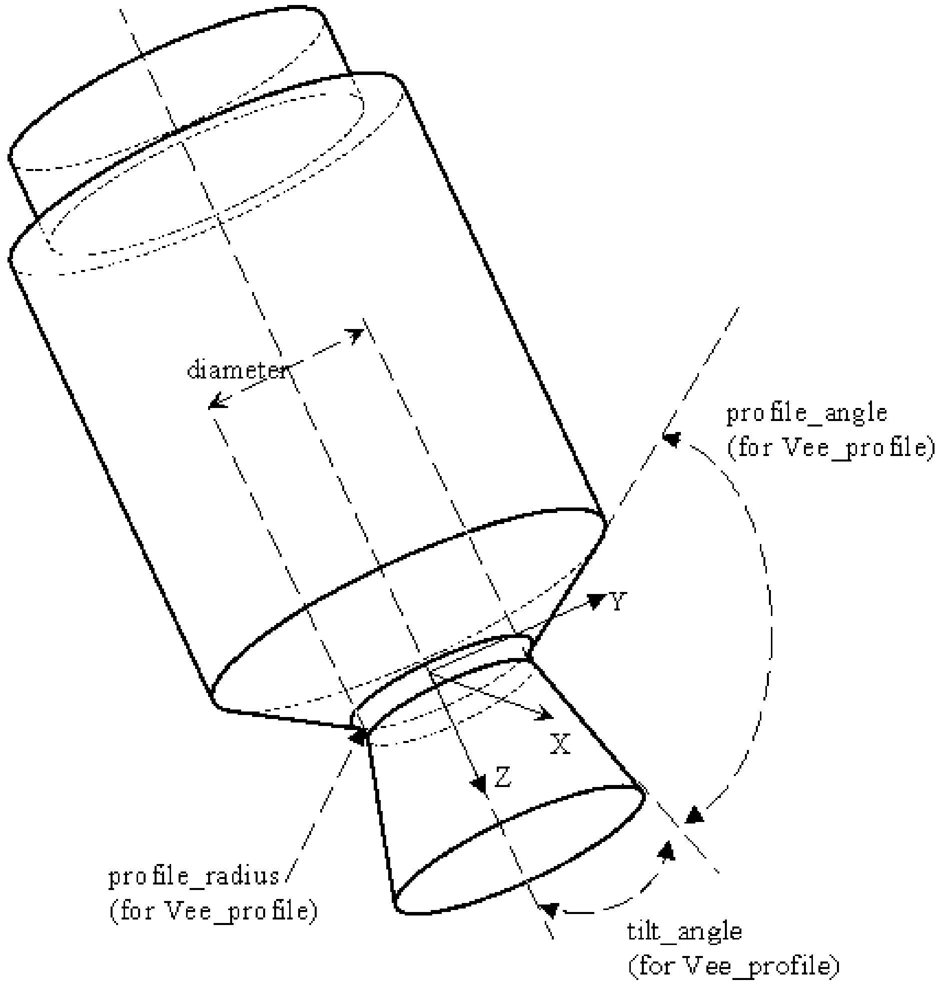


Figure 80 — Outer_diameter_to_shoulder

4.2.152.2 feature_length

The length specifies the size of a Outer_diameter feature_to_shoulder, measured along the feature's axis. See 4.3.169 for the application assertion.

4.2.152.3 v_shape_boundary

The v_shape_boundary specifies an outline or shape that shall be revolved about an axis. The Vee_profile specifies the revolved shape required by an Outer_diameter_to_shoulder. The placement of the profile shall be along the X-axis of the Outer_diameter_to_shoulder at a specified distance away from the origin. The orientation of the Y-axis of the Vee_profile shall be the same as the Y-axis of the Outer_-diameter_to_shoulder and the X-axis of the Vee_profile shall be the same as the Z-axis of the Outer_-diameter_to_shoulder. See 4.3.170 for the application assertion.

4.2.153 Outer_round

An Outer_round is a type of Machining_feature (see 4.2.126) that is an outline or significant shape that is swept through a complete revolution about an axis. Each Outer_round is either an Outer_diameter (see 4.2.151) or an Outer_diameter_to_shoulder (see 4.2.152). The axis of revolution shall be the same as the Z-axis of the feature.

4.2.154 Parallelism_tolerance

A Parallelism_tolerance is a type of Geometric_tolerance (see 4.2.104) that is the condition of a surface equidistant at all points from a datum plane or an axis equidistant along its length to a datum axis or plane. A Parallelism_tolerance specifies one of the following:

— A tolerance zone defined by two planes or lines parallel to a datum plane, or axis, within which the line elements of the surface or axis shall lie.

— A cylindrical tolerance zone whose axis is parallel to a datum axis within which the axis shall lie

NOTE 1 Figure 81 illustrates Parallelism_tolerance for a plane.

NOTE 2 Figure 82 illustrates Parallelism_tolerance for an axis.

NOTE 3 The Parallelism_tolerance definition is derived from ISO 1101:2004, 18.9.

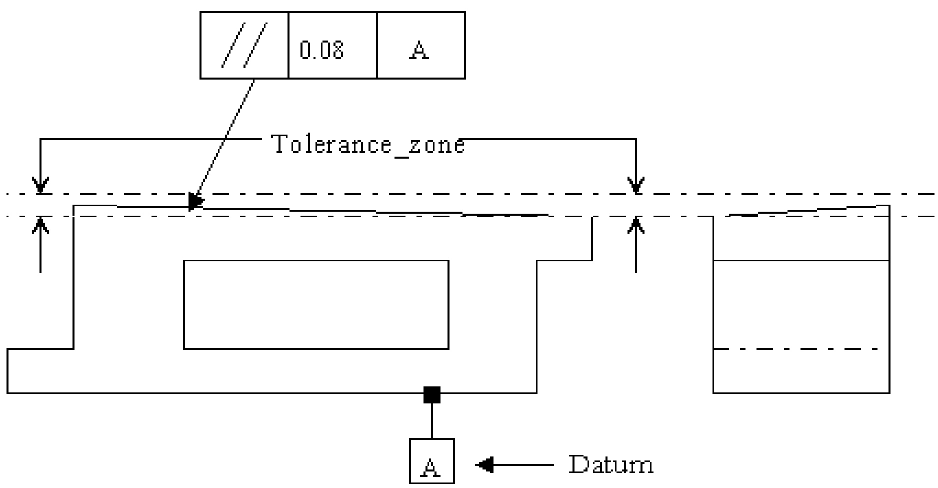


Figure 81 — Parallelism_tolerance for a plane

The data associated with a Parallelism_tolerance are the following:

— affected_plane;

— geometric_reference;

— segment_size.

4.2.154.1 affected_plane

The affected_plane specifies the plane to apply the tolerance value. The affected_plane is equivalent to a 2D drawing view. The affected_plane need not be specified for a particular Parallelism_tolerance. See 4.3.172 for the application assertion.

4.2.154.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.171 for the application assertion.

4.2.154.3 segment_size

The segment_size specifies the length of a surface to apply a tolerance if the Parallelism_tolerance is not applied to the total length. A segment_size need not be specified for a particular Parallelism_tolerance.

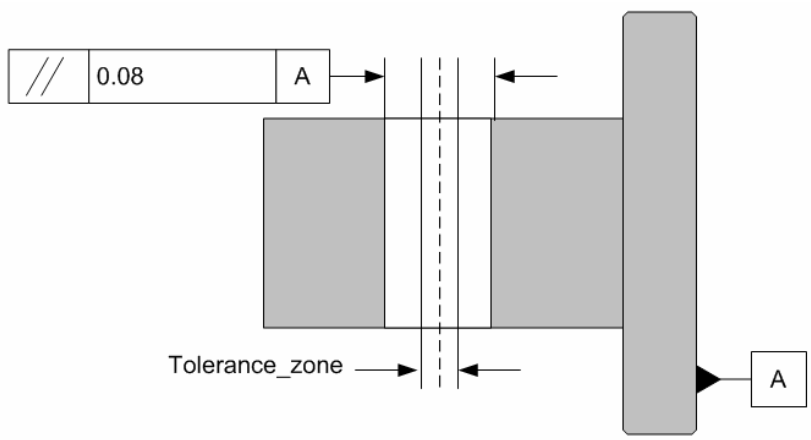


Figure 82 — Parallelism_tolerance for an axis

4.2.155 Part

A Part is the physical item which is intended to be produced through the manufacturing process. Each Part may be one of the following: a Manufactured_assembly (see 4.2.127), or a Single_piece_part (see 4.2.222). The data associated with a Part are the following:

— manufacture_authorization;

— manufactured_by_organization;

— manufactured_by_person;

— owned_by_organization;

— owned_by_person;

— part_description;

— part_id;

— part_name;

— part_revision_id;

— physical_form;

— property_characteristics;

— quantity_ordered;

— security_classification.

4.2.155.1 manufacture_authorization

The manufacture_authorization specifies the approval description for the manufacture of a Part. The manufacture_authorization need not be specified for a particular Part. There may be more than one manufacture_authorization for a Part. See 4.3.173 for the application assertion.

4.2.155.2 manufactured_by_organization

The manufactured_by_organization specifies the organization responsible for the manufacture of the Part. There may be more than one manufactured_by_organization for a Part. See 4.3.175 for the application assertion.

4.2.155.3 manufactured_by_person

The manufactured_by_person specifies the person responsible for the manufacture of the Part. There may be more than one manufactured_by_person for a Part. See 4.3.176 for the application assertion.

4.2.155.4 owned_by_organization

The owned_by_organization specifies the organization that will own the completed manufactured Part There may be more than one owned_by_organization for a Part. See 4.3.175 for the application assertion.

4.2.155.5 owned_by_person

The owned_by_person specifies the person that will own the completed manufactured Part. There may be more than one owned_by_person for a Part. See 4.3.176 for the application assertion.

4.2.155.6 part_description

The part_description specifies in a textural format, human interpretable summary of the Part's characteristics which is appropriate for the manufacturing of the part. Any comments or requirements on part characteristics may be given as a portion of the part_description.

4.2.155.7 part_id

The part_id specifies a unique identifier for a Part within an organization.

4.2.155.8 part_name

The part_name specifies a word or group of words by which a Part commonly is called within an organization. This name need not be unique within an organization.

4.2.155.9 part_revision_id

The part_revision_id specifies a unique identifier that defines the appropriate level of change that is incorporated into the design of the Part for the manufacturing function. The part_revision_id need not specify the latest version of a Part's design to be manufactured.

4.2.155.10 physical_form

The physical_form specifies the shape of the Part. See 4.3.178 for the application assertion.

4.2.155.11 property_characteristic

The property_characteristic specifies the associated data about the physical structure (see 4.2.186). The property characteristic need not be specified for a particular Part. There may be more than one property characteristic for a Part. See 4.3.177 for the application assertion.

4.2.155.12 quantity_ordered

The quantity_ordered specifies the number of Parts to be manufactured. There may be more than one quantity_ordered for a Part. See 4.3.174 for the application assertion.

4.2.155.13 security_classification

The security_classification specifies an organizational, national or international code that defines the availability of the Part or information about the Part to a particular individual, group of individuals, organization or group of organizations.

4.2.156 Part_dimensioning_standard

A Part_dimensioning_standard is a type of document assignment 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;

— assigned_document.

4.2.156.1 applied_part

The applied_part specifies the part that uses the dimensioning standard. See 4.3.179 for the application assertion.

4.2.156.2 assigned_document

The assigned_document specifies the Document_file (see 4.3.180) that is used to provide information. See 4.3.179 for the application assertion.

4.2.157 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.

4.2.157.1 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 4.3.181 for the application assertion.

4.2.157.2 oriented_physical_form

The oriented_physical_form specifies the shape of a part or sub-assembly that is being re-positioned. See 4.3.182 for the application assertion.

4.2.157.3 resulting_orientation

The resulting_orientation specifies the orientation of a part or sub-assembly in an assembly or another sub_assembly. See 4.3.181 for the application assertion.

4.2.158 Part_property

A Part_property is a specific characteristic about the form, fit, or function of a part. The data associated with a Part_property are the following:

— property_characteristic.

4.2.158.1 property_characteristic

The property_characteristic specifies the parameter to describe the Part_property. The property_-characteristic need not be specified for a particular Part_property. There may be more than one property_characteristic for a Part_property. See 4.3.183 for the application assertion.

4.2.159 Partial_area_definition

A Partial_area_definition is the limitations of a surface for applying a Machining_feature. Thread and Knurl features are applied to cylindrical shapes. Partial_area_definition places a limitation on how much and where to apply the feature on the cylindrical shape.

NOTE Figure 14 and Figure 38 illustrates a Defined_thread with a Partial_area_definition. The drawing call out '5.12 MIN.THREAD' defines the amount of cylindrical shape that has the thread applied.

The data associated with a Partial_area_definition are the following:

— effective_length;

— maximum_length;

— placement.

4.2.159.1 effective_length

The effective_length specifies the length of the thread which is usable by the feature. When applied to a knurl the effective_length defines the overall length of the knurl. The effective_length is not required for a Knurl feature. See 4.3.184 for the application assertion.

4.2.159.2 maximum_length

The maximum_length specifies the dimension along a surface to apply a feature. A Thread is a type of Machining_features that is applied to a surface. The dimensional distance limits the length along the surface axis for defining these feature objects. A knurl does not require maximum_length. The maximum_length need not be specified for a particular Partial_area_definition. See 4.3.184 for the application assertion.

4.2.159.3 placement

The placement specifies where to locate the Partial_area_definition. See 4.3.185 for the application assertion.

4.2.160 Partial_circular_path

A Partial_circular_path is a type of Circular_path (see 4.2.30) that is a direction of travel along an arc of constant radius around an axis. The path shall begin and end at different points on the arc.

NOTE Figure 83 illustrates two Slot feature with a Square_U_profile and a Partial_circular_paths.

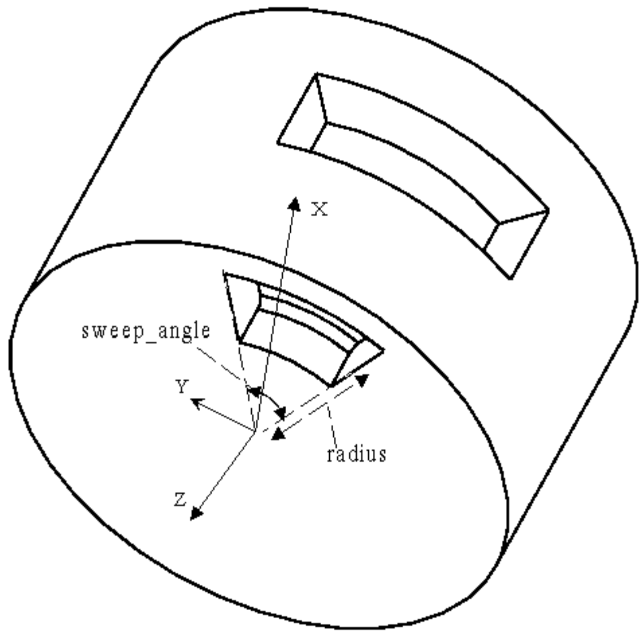


Figure 83 — Partial_circular_path

The data associated with a Partial_circular_path are the following:

— sweep_angle.

4.2.160.1 sweep_angle

The sweep_angle specifies the size of the angle to define an arc shaped path. See 4.3.186 for the application assertion.

4.2.161 Partial_circular_profile

A Partial_circular_profile is a type of Open_profile (see 4.2.145) that is specified by an arc. The arc shall be a constant radius swept about a point. The orientation of the profile shall be positioned at the origin of the arc, with one end point of the arc on the X-axis.

NOTE Figure 84 illustrates a Partial_circular_profile.

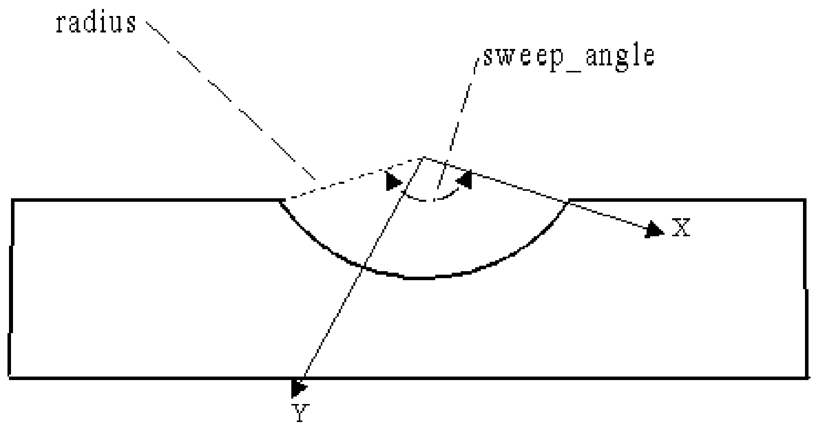


Figure 84 — Partial_circular_profile

The data associated with a Partial_circular_profile are the following:

— radius;

— sweep_angle.

4.2.161.1 radius

The radius specifies the size of the arc to define a Partial_circular_profile. See 4.3.187 for the application assertion.

4.2.161.2 sweep_angle

The sweep_angle specifies the size of the angle to define an circular shaped profile. See 4.3.187 for the application assertion.

4.2.162 Partial_circular_shape_profile

A Partial_circular_shape_profile is a type of Shape_profile (see 4.2.220) that defines a volume that is not enclosed on all sides. The data associated with a Partial_circular_shape_profile are the following:

— open_boundary.

4.2.162.1 open_boundary

The open_boundary specifies the outline of the Shape_profile feature. The outline defines an area that shall be circular and shall not be enclosed. The placement of the open_boundary shall be with the origin of the Path, that defines the profile, at the origin of the Partial_circular_shape_profile The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Partial_circular_shape_profile. See 4.3.188 for the application assertions.

4.2.163 Path

A Path is a continuous set of curves that define a direction of travel. These curves do not intersect or duplicate themselves. A Path shall have its own orientation which may be the same orientation as the Machining_feature which requires it as a part of a feature definition. The orientation of a Path may be the same orientation as the Machining_feature. Each Path is either a Circular_path (see 4.2.30), a General_-path (see 4.2.94), or a Linear_path (see 4.2.118). The data associated with a Path are the following:

— placement.

4.2.163.1 placement

The placement specifies where to locate the Path. See 4.3.189 for the application assertion.

4.2.164 Path_element

A Path_element is a type of Shape_element (see 4.2.219) that is a continuous set of geometric curve that represent the path for a particular Machining_feature.

4.2.165 Pedigree_creation_order

A Pedigree_creation_order is a document that defines the requirements for data to be gathered, documented and archived about a particular instance of a Part or batch of Parts throughout the manufacturing process. The data associated with a Pedigree_creation_order are the following:

— order_id.

4.2.165.1 order_id

The order_id specifies a unique identifier for the Pedigree_creation_order.

4.2.166 Perpendicularity_tolerance

A Perpendicularity_tolerance is a type of Geometric_tolerance (see 4.2.104) that is the amount of deviation of a surface to a perpendicular to a datum. The actual surface shall lie within two planes a tolerance apart and perpendicular to a datum surface.

NOTE 1 Figure 85 and Figure 86 illustrates a Perpendicularity_tolerance.

NOTE 2 The Perpendicularity_tolerance definition is derived from ISO 1101:2004, 18.10.

The data associated with a Perpendicularity_tolerance are the following:

— affected_plane;

— geometric_reference;

— segment_size.

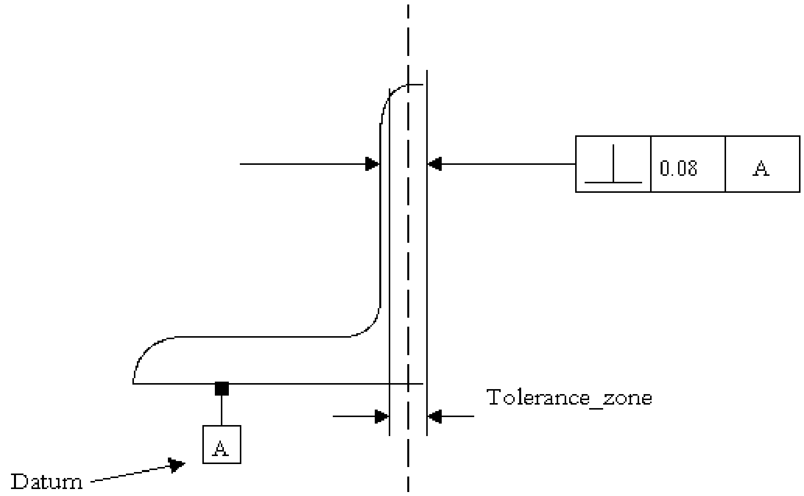


Figure 85 — Perpendicularity_tolerance for a plane

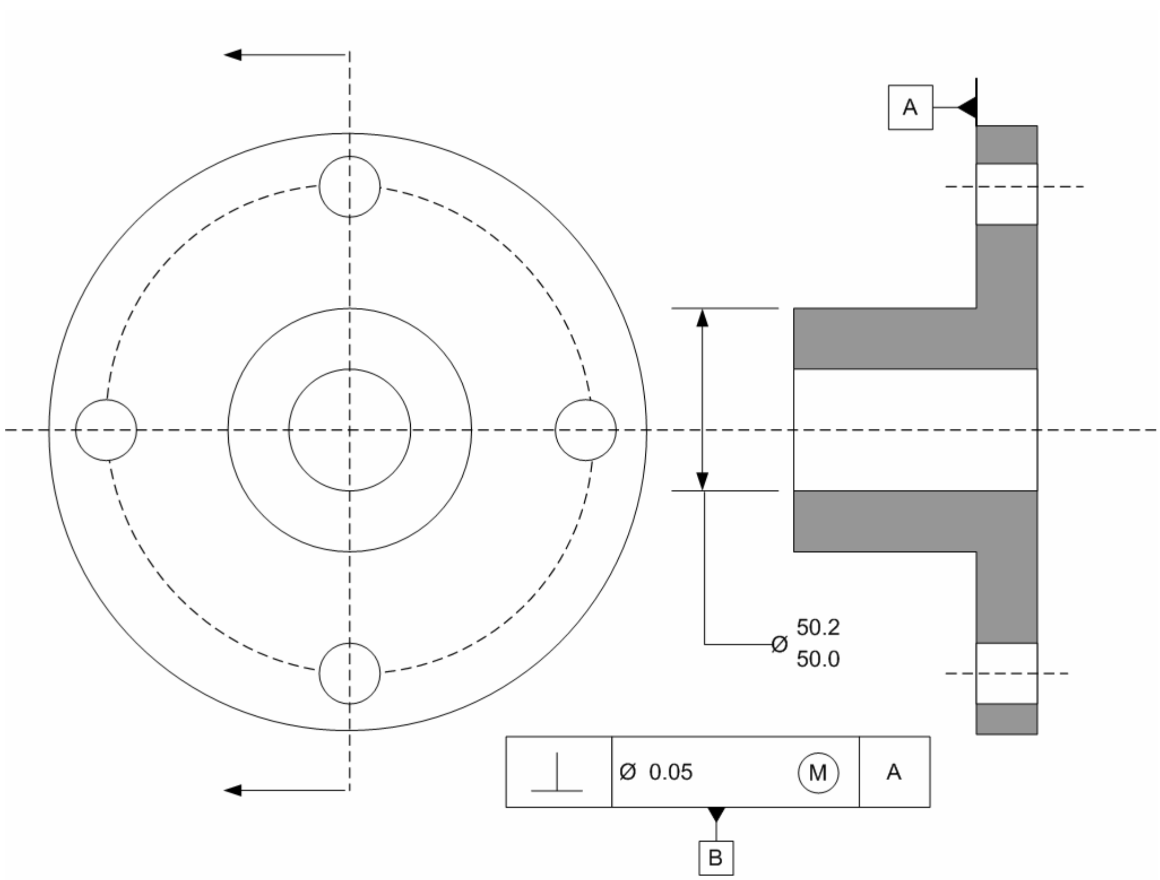


Figure 86 — Perpendicularity_tolerance for a axis

4.2.166.1 affected_plane

The affected_plane specifies the plane to apply the tolerance value. The affected_plane is equivalent to a 2D drawing view. The affected_plane need not be specified for a particular Perpendicularity_tolerance. See 4.3.191 for the application assertion.

4.2.166.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.190 for the application assertion.

4.2.166.3 segment_size

The segment_size specifies the length of a surface to apply a tolerance if the Perpendicularity_tolerance is not applied to the total length. The segment_size need not be specified for a particular Perpendicularity_-tolerance.

4.2.167 Person

A Person is an individual human being. The data associated with a Person are the following:

— person_address;

— person_id;

— person_name;

— person_phone_number.

4.2.167.1 person_address

The person_address specifies a location at which a Person is contacted. This location may be physical or electronic.

4.2.167.2 person_id

The person_id specifies the unique identification of an individual within the context to which that individual is being referenced.

EXAMPLE Within the United States individuals have unique social security numbers. Within a particular organization employing individuals, each individual may have a unique employee id.

4.2.167.3 person_name

The person_name specifies a word or group of words by which a Person is commonly referred. The person_name may be comprised of compound elements that are commonly referred to as first name, last name, middle names and titles or suffixes.

4.2.167.4 person_phone_number

The person_phone_number specifies the telephone number at which the Person can be reached.

4.2.168 Person_in_organization

A Person_in_organization is a person within the context of an organization. The data associated with a Person_in_organization are the following:

— company;

— employee;

— role.

4.2.168.1 company

The company specifies the functional group that employs the Person. See 4.3.192 for the application assertion.

4.2.168.2 employee

The employee specifies Person employed by the Organization. See 4.3.193 for the application assertion.

4.2.168.3 role

The role specifies the purpose that the person is fulfilling in an organization.

4.2.169 Placed_target

A Placed_target is a type of Datum_target (see 4.2.56) that is the implicit definition of a Datum_target. A Placed_target is either a Target_point (see 4.2.243), a Target_line (see 4.2.242), a Target_rectangle (see 4.2.244), or a Target_circle (see 4.2.241). The data associated with Placed_target are the following:

— placement.

4.2.169.1 placement

The placement specifies location and orientation for the implicit definitions of the types of Placed_target. See 4.3.194 for the application assertion.

4.2.170 Planar_element

A Planar_element is a type of Shape_element (see 4.2.219) that is a flat surface. The data associated with a Planar_element are the following:

— location;

— normal.

4.2.170.1 location

The location specifies the position of the planar surface. See 4.3.196 for the application assertion.

4.2.170.2 normal

The normal specifies the vector which indicates the normal of a plane being defined for the planar surface. See 4.3.195 for the application assertion.

4.2.171 Planar_face

A Planar_face is a type of Multi_axis_feature(see 4.2.140) that is an unbounded planar cut of a part. The Planar_face shall have an orientation such that the Z-axis is the direction away from the part.

NOTE Figure 87 illustrates the Planar_face.

The data associated with a Planar_face are the following:

— course_of_travel;

— face_boundary;

— removal_boundary;

— removal_depth;

— removal_direction;

— volume_not_removed.

4.2.171.1 course_of_travel

The course_of_travel specifies a straight line with magnitude and direction. The placement and orientation of the Linear_path shall be the same as the Planar_face feature. See 4.3.199 for the application assertion.

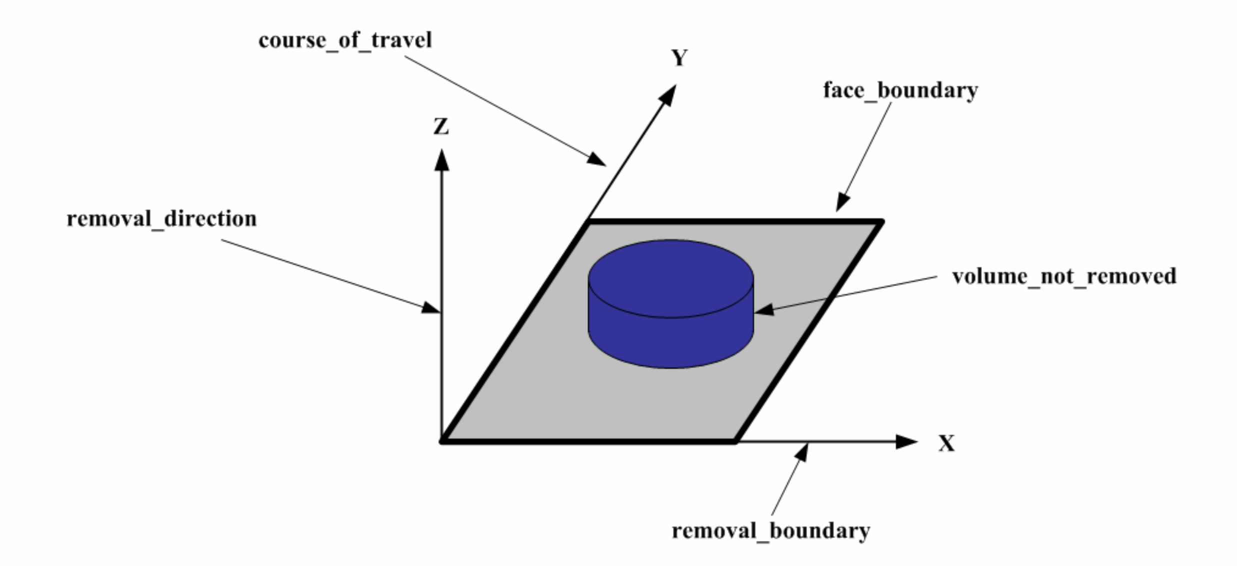


Figure 87 — Planar_face

4.2.171.2 face_boundary

The face_boundary specifies the complete or partial outside final shape of the part after the planar cut has been applied. A Planar_face may but need not require face_boundary to be defined. See 4.3.197 for the application assertion.

4.2.171.3 removal_boundary

The removal boundary specifies a line with direction and magnitude that when swept along a path defines the area on a part for volume removal. The orientation and placement of the Linear_profile shall be the same as the Planar_face. See 4.3.200 for the application assertion.

4.2.171.4 removal_depth

The removal_depth specifies a measured distance from the bottom of a point that is outside of the planar_face. Removal_depth places a limitation on the Planar_face definition so not to interfere with other features that might be nearby. The placement and orientation of the Linear_path that defines removal_depth shall be the same as the planar_face. See 4.3.202 for the application assertion.

EXAMPLE If a portion of the part should extend over the top of the planar_face, the depth value would not interfere with it.

4.2.171.5 removal_direction

The removal_direction specifies the direction of material removal from the Planar_face feature. See 4.3.198 for the application assertion.

4.2.171.6 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the planar_face. The Boss or Protrusion feature define the shape of the material that is to remain in the planar_face. See 4.3.201 or 4.3.203 for the application assertion.

4.2.172 Planar_pocket_bottom_condition

A Planar_pocket_bottom_condition is a type of Pocket_bottom_condition (see 4.2.178) that characterizes the bottom of a pocket which is flat.

NOTE Figure 98 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition. Figure 94 illustrates a Rectangular_closed_pocket with a Planar_pocket_bottom_condition.

The data associated with a Planar_pocket_bottom_condition are the following:

— floor_location;

— floor_normal;

— floor_radius.

4.2.172.1 floor_location

The floor_location specifies the position of the bottom of a pocket feature for a planar floor. See 4.3.205 for the application assertion.

NOTE Figure 99 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition and a floor_location.

4.2.172.2 floor_normal

The floor_normal specifies the vector which indicates the normal of a plane being defined for the bottom of a pocket. See 4.3.204 for the application assertion.

NOTE Figure 99 illustrates the Rectangular_open_pocket with a Planar_pocket_bottom_condition and a floor_normal.

4.2.172.3 floor_radius

The floor_radius specifies the amount of curvature for an arc between the bottom and the sides of a Pocket feature. See 4.3.206 for the application assertion.

4.2.173 Planar_profile_floor

A Planar_profile_floor is a type of Profile_floor (see 4.2.183) that characterizes the bottom of a Shape_profile feature which is flat.

NOTE Figure 107 illustrates a Shape_profile with a Planar_profile_floor.

The data associated with a Planar_profile_floor are the following:

— floor.

4.2.173.1 floor

A floor specifies a planar floor for the Shape_profile (see 4.2.220) feature. The normal to the plane shall be in the direction away from the Planar_profile_floor feature. See 4.3.207 for the application assertion.

4.2.174 Planar_rib_top_floor

A Planar_rib_top_floor is a type of Rib_top_floor (see 4.2.211) that is the bottom of a Rib_top (see 4.2.210) which is flat. The data associated with a Planar_rib_top_floor are the following:

— boundary;

— floor_face.

4.2.174.1 boundary

The boundary specifies the complete or partial outside final shape of the Rib_top feature. See 4.3.208 for the application assertion.

4.2.174.2 floor_face

A floor_face specifies a planar floor for the Rib_top (see 4.2.210) feature. The normal to the plane shall be in the direction away from the Rib_top feature. See 4.3.209 for the application assertion.

4.2.175 Planar_top_condition

A Planar_top_condition is a type of Boss_top_condition (see 4.2.12) for a Boss that is flat.

NOTE Figure 16 illustrates a Circular_boss with a Planar_top_condition.

The data associated with a Planar_top_condition are the following:

— top_location;

— top_normal.

4.2.175.1 top_location

The top_location specifies the position of the top of a boss feature for a planar top. See 4.3.211 for the application assertion.

4.2.175.2 top_normal

The top_normal specifies the vector that indicates the normal of a plane being defined for the top of a boss. See 4.3.210 for the application assertion.

4.2.176 Plus_minus_value

The Plus_minus_values is the upper and lower limits or tolerance value applied directly to a dimension. When applied to a Dimensional_tolerance, the dimensional_value is the tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value is the tolerance value.

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.

4.2.176.1 lower_limit

The lower_limit specifies the low limit value.

4.2.176.2 significant_digits

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

4.2.176.3 upper_limit

The upper_limit specifies the high limit value.

4.2.177 Pocket

A Pocket is a type of Multi_axis_feature(see 4.2.140) that is a volume with a specific shape, removed from the part. The sides of a pocket may be parallel to the pocket's orientation vector coming out of the pocket or the sides may be tapered. The placement may be at the bottom of the Pocket with the Z-axis in the direction out of the pocket or at the top of the pocket with the Z-axis in the direction into the pocket. Each Pocket is either a Cutout (see 4.2.50), a General_pocket (see 4.2.96), a Recess (see 4.2.193), a Rectangular_closed_pocket (see 4.2.195), or a Rectangular_open_pocket (see 4.2.200).

The data associated with a Pocket are the following:

— base_radius;

— bottom_condition;

— change_in_boundary;

— pocket_depth.

4.2.177.1 base_radius

The base_radius specifies a radius shape blend between a Pocket and the surrounding Part surface at the top of the Pocket. See 4.3.215 for the application assertion.

4.2.177.2 bottom_condition

The bottom_condition specifies the shape of the bottom of a Pocket feature. See 4.3.216 and 4.3.217 for the application assertions.

4.2.177.3 change_in_boundary

The change_in_boundary specifies a taper that defines the change in shape of the Pocket. The change_-in_boundary need not be specified for a particular Pocket. See 4.3.212 and 4.3.213 for the application assertion.

4.2.177.4 pocket_depth

The pocket_depth specifies an measured distance from the bottom of a pocket to a point that is outside of the pocket feature. Pocket_depth places a limitation on the Pocket definition so not to interfere with other features that might be nearby. The placement and orientation of the Linear_path that defines pocket_-depth shall be the same as the Pocket feature. See 4.3.214 for the application assertion.

EXAMPLE If a portion of the part should extend over the top of the pocket feature, the depth value would not interfere with it.

4.2.178 Pocket_bottom_condition

A Pocket_bottom_condition specifies the bottom state for a pocket. The Pocket bottom may be flat, or any arbitrary shape, or the pocket may pass through the part. Each Pocket_bottom_condition is either a a General_pocket_bottom_condition (see 4.2.97), a Planar_pocket_bottom_condition (see 4.2.172), or a Through_pocket_bottom_condition (see 4.2.250).

The data associated with a Pocket_bottom_condition are the following:

— start_or_end.

4.2.178.1 start_or_end

The start_or_end specifies a boolean value of TRUE if the Pocket_bottom_condition is positioned at the start of the of the defining Linear_path (see 4.2.118) for the Pocket, and a value of FALSE if it is at the end of the of the defining Linear_path for the Pocket.

NOTE Figure 88 illustrates the start_or_end attribute for a Rectangular_open_pocket

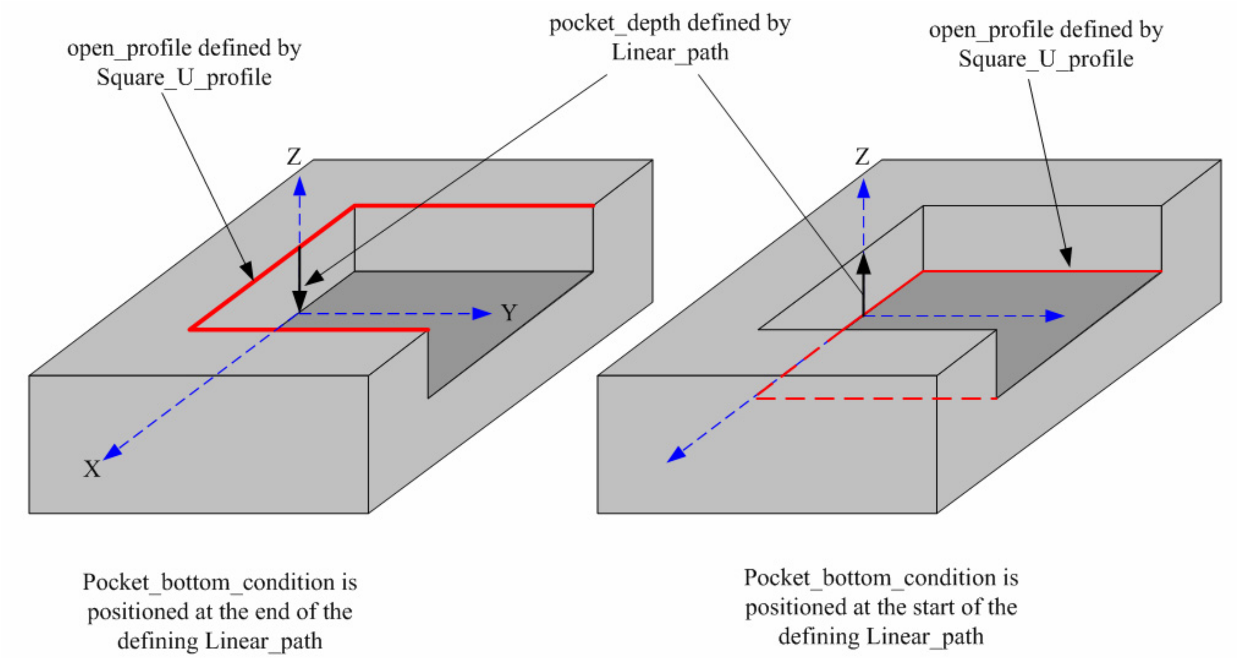


Figure 88 — Planar_pocket_bottom_condition

4.2.179 Position_tolerance

A Position_tolerance is a type of Geometric_tolerance (see 4.2.104) that denotes a tolerance zone for a theoretically exact position_tolerance of a surface, and is established with respect to a datum.

NOTE 1 Figure 89 illustrates the Position_tolerance.

NOTE 2 The Position_tolerance definition is derived fromISO 1101:2004, 18.12.

The data associated with a Position_tolerance are the following:

— affected_plane;

— geometric_reference;

— value_qualifier.

4.2.179.1 affected_plane

The affected_plane specifies the plane to apply the tolerance value. The affected_plane is equivalent to a 2D drawing view. The affected_plane need not be specified for a particular Position_tolerance. See 4.3.219 for the application assertion.

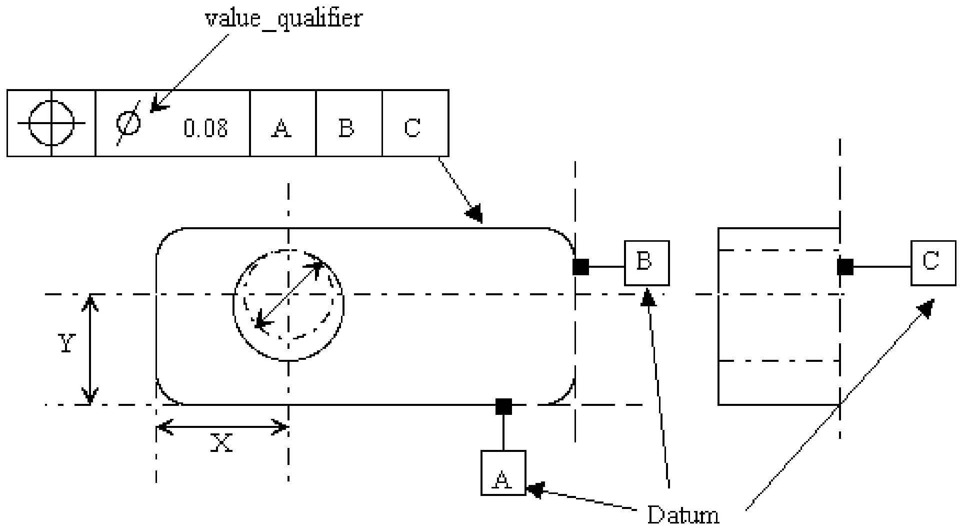


Figure 89 — Position_tolerance

4.2.179.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.218 for the application assertion.

4.2.179.3 value_qualifier

The value_qualifier specifies the type of tolerance. A Position_tolerance of TRUE is a diametric tolerance. A Position_tolerance of FALSE is not a diametric tolerance.

4.2.180 Process_property

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

EXAMPLE Examples of Process_property may be paint or coat.

The data associated with a Process_property are the following:

— process_name;

— property_characteristic.

4.2.180.1 process_name

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

4.2.180.2 property_characteristic

The property_characteristic specifies the parameter to describe the Process_property. The property_-characteristic need not be specified for a particular Process_property. There may be more than one property_characteristic for a Process_property. See 4.3.220 for the application assertion.

4.2.181 Profile

A Profile is a planar outline used in the definition of a feature. A Profile may be either open or closed. A Profile shall be in the X-Y plane and have an orientation that will position it in reference to a Machining_feature, which may require a profile as a part of its definition. Each Profile is either a Closed_profile (see 4.2.35) or an Open_profile (see 4.2.145). The data associated with a Profile are the following:

— placement.

4.2.181.1 placement

The placement specifies where to locate the Profile in reference to the Part orientation. See 4.3.221 for the application assertion.

4.2.182 Profile_feature

A Profile_feature is a type of Multi_axis_feature (see 4.2.140) that is the removal of excess material from the boundary shape of a part. Each Profile_feature is either a General_outside_profile (see 4.2.93), or a Shape_profile(see 4.2.220). The data associated with a Profile_feature are the following:

— profile_swept_shape.

4.2.182.1 profile_swept_shape

The profile_swept_shape specifies an implicit 2D line (see 4.2.119) definition that, when combined with a Profile, creates the shape of the Profile_feature. The profile_swept_shape places a limitation on the Profile_feature definition so not to interfere with other features that might be nearby. The placement of the Linear_path that defines profile_swept_shape shall be the same as the Profile_feature feature. The orientation shall be with the Z-axis toward the direction of travel of the profile boundary, and the Y-axis in the direction away from the part material. See 4.3.222 for the application assertion.

EXAMPLE If a portion of the part should extend over the top of the Profile_feature, the depth value would not interfere with it.

4.2.183 Profile_floor

A Profile_floor is the bottom condition for a Shape_profile (see 4.2.220). The Profile_floor may be flat, or any arbitrary shape. Each Profile_floor is either a General_profile_floor (see 4.2.98) or a Planar_profile_floor (see 4.2.173).

The data associated with a Profile_floor are the following:

— floor_radius;

— start_or_end.

4.2.183.1 floor_radius

The floor_radius specifies the radius of curvature for an arc between the bottom and the sides of a Pocket feature. See 4.3.223 for the application assertion.

4.2.183.2 start_or_end

The start_or_end specifies a boolean value of TRUE if the Profile_floor is positioned at the end of a Shape_profile, and a value of FALSE if it is at the start of the Shape_profile.

4.2.184 Project_order

A Project_order is a document that is used to initiate and track a project within the manufacturing organization. A Project_order may track a single part or a part batch. The data associated with a Project_order are the following:

— ordered_resource;

— part_status;

— pedigree_creation_status;

— project_order_id;

— release_authorization;

— resource_acquisition_status;

— shop_work_status;

— technical_data_package_status.

4.2.184.1 ordered_resource

The ordered_resource specifies the resources required by the Project_order. There may be more than one ordered_resource for a Project_order. See 4.3.228 for the application assertion.

4.2.184.2 part_status

The part_status specifies Part that is tracked by the Project_order. There may be more than one part_status for a Project_order. See 4.3.226 for the application assertion.

4.2.184.3 pedigree_creation_status

The pedigree_creation_status specifies the Pedigree_creation_order that is tracked by the Project_order. The pedigree_creation_status need not be specified for a particular Project_order. See 4.3.227 for the application assertion.

4.2.184.4 project_order_id

The project_order_id specifies a unique identifier for the Project_order.

4.2.184.5 release_authorization

The release_authorization specifies the approval to manufacture a Part. See 4.3.224 for the application assertion.

4.2.184.6 resource_acquisition_status

The resource_acquisition_status specifies Resource_acquisition_order that is tracked by the Project_-order. The resource_acquisition_status need not be specified for a particular Project_order. See 4.3.229 for the application assertion.

4.2.184.7 shop_work_status

The shop_work_status specifies the shop_work_order that is tracked by the Project_order. The shop_-work_status need not be specified for a particular Project_order. See 4.3.230 for the application assertion.

4.2.184.8 technical_data_package_status

The technical_data_package_status specifies the Digital_technical_data_package_work_order that is tracked by the Project_order. The technical_data_package_status need not be specified for a particular Project_order. See 4.3.225 for the application assertion.

4.2.185 Projection

A Projection is an extension of a feature from the part so that a tolerance zone can be created. A feature is extended from one end of a feature for a specified length. The data associated with a Projection are the following:

— projection_end;

— projection_length.

4.2.185.1 projection_end

The projection_end specifies the physical shape that is the extension of a feature. See 4.3.231 for the application assertion.

4.2.185.2 projection_length

The projection_length specifies the amount to extend the end of a feature.

4.2.186 Property

A Property is a characteristic associated with the physical structure or integrity of an element of a part. The data associated with a Property are the following:

— material_characteristic;

— property_name;

— part_property_characteristic;

— process_characteristic;

— property_characteristic;

— property_description;

— surface_characteristic.

4.2.186.1 material_characteristic

The material_characteristic specifies the information that describe material for manufacturing the Part. The material_characteristic need not be specified for a particular Property. There may be more than one material_characteristic for a Property. See 4.3.232 for the application assertion.

4.2.186.2 property_name

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

4.2.186.3 part_property_characteristic

The part_property_characteristic specifies the information that describe properties of the Part. The part_-property_characteristic need not be specified for a particular Property. There may be more than one part_property_characteristic for a Property. See 4.3.233 for the application assertion.

4.2.186.4 process_characteristic

The process_characteristic specifies information that describe processes for manufacturing the part. The process_characteristic need not be specified for a particular Property. There may be more than one process_characteristic for a Property. See 4.3.234 for the application assertion.

4.2.186.5 property_characteristic

The property_characteristic specifies information that describe properties of the Part. The property_-characteristic need not be specified for a particular Property. There may be more than one property_-characteristic for a Property. See 4.3.235 for the application assertion.

4.2.186.6 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 4.3.236 for the application assertion.

4.2.186.7 surface_characteristic

The surface_characteristic specifies information that describe surface conditions of the Part. The surface_characteristic need not be specified for a particular Property. There may be more than one surface_characteristic for a Property. See 4.3.237 for the application assertion.

4.2.187 Property_BSU

A Property_BSU is a type of BSU (see 4.2.17) that identifies a property basic semantical unit of a class in a parts library.

The data associated with a Property_BSU are the following:

— name_scope.

4.2.187.1 name_scope

The name_scope specifies the class this property belongs to. See 4.3.238 for the application assertion.

4.2.187.2 version

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

4.2.188 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: a Descriptive_parameter (see 4.2.62) or a Numeric_parameter (see 4.2.143). The data associated with a Property_parameter are the following:

— parameter_name.

4.2.188.1 parameter_name

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

4.2.189 Property_value

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

The data associated with a Property_value are the following:

— property_BSU;

— value_amount.

4.2.189.1 property_BSU

The property_BSU specifies the Propert_BSU (see 4.2.187) that defines the basic semantical unit. See 4.3.239 for the application assertion.

4.2.189.2 value_amount

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

4.2.190 Protrusion

A Protrusion is a type of Multi_axis_feature(see 4.2.140) that is an arbitrary shape that extends out from a surrounding surface.

NOTE Figure 90 illustrates a Protrusion.

The data associated with a Protrusion are the following:

— shape_volume.

4.2.190.1 shape_volume

The shape_volume specifies the arbitrary shape that defines the shape of a Protrusion. See 4.3.240 for the application assertion.

4.2.191 Radial_dimension_tolerance

A Radial_dimension_tolerance is a type of Size_tolerance (see 4.2.223) that is the allowable variation for the radial distance from the center of a circular curve to a point on the curve.

NOTE Figure 91 illustrates the Radial_dimension_tolerance.

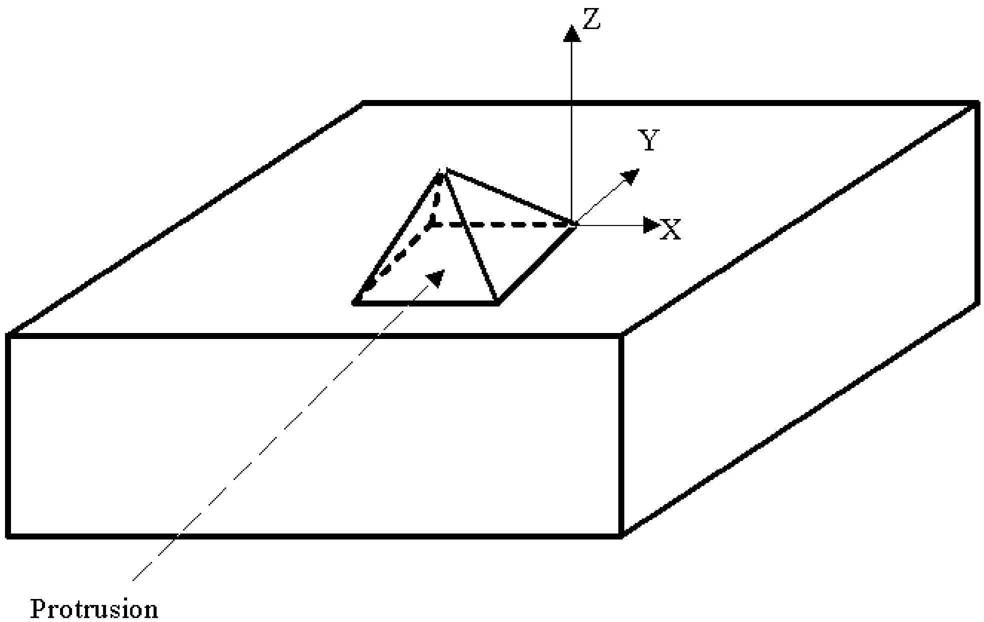


Figure 90 — Protrusion

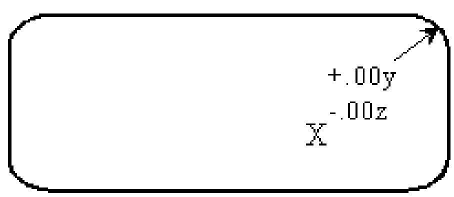


Figure 91 — Radial_dimension_tolerance

4.2.192 Radiused_slot_end_type

A Radiused_slot_end_type is a type of Slot_end_type (see 4.2.225) that is an end condition of a slot that shall be a cylindrical shape tangent to both of the adjacent Slot wall surfaces.

NOTE Figure 92 illustrates the Radiused_slot_end_type.

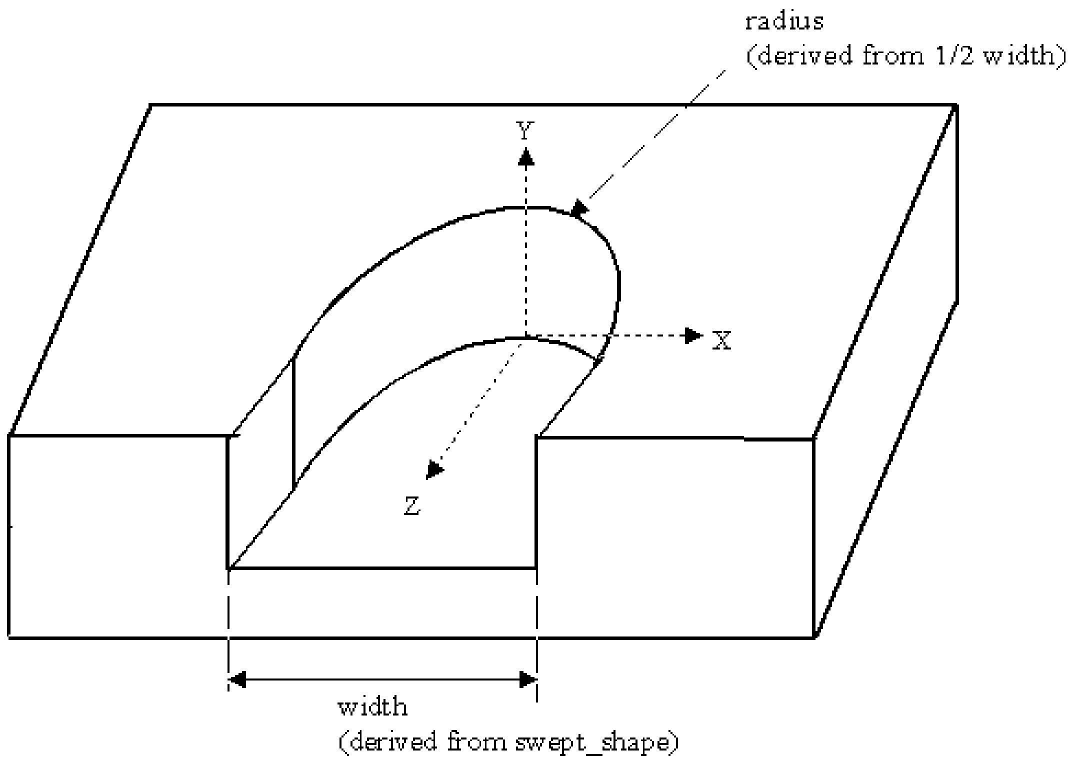


Figure 92 — Radiused_slot_end_type

4.2.193 Recess

A Recess is a type of a Pocket (see 4.2.177) that has a bottom, and a floor radius, but the sides of the pocket shall be no higher that the floor radius. The Recess shall not pass entirely through the part but shall have a floor bottom. The type of bottom condition shall be a Pocket bottom_condition (see 4.2.178).

EXAMPLE Figure 93 illustrates the Recess. A Recess is commonly found in the bottom of a Pocket feature.

The data associated with a Recess are the following:

— bottom_condition;

— fillet_boundary;

— volume_not_removed.

4.2.193.1 bottom_condition

The bottom_condition specifies the shape of the bottom of a Recess feature. The bottom_condition shall not pass entirely through the part. See 4.3.242 for the application assertion.

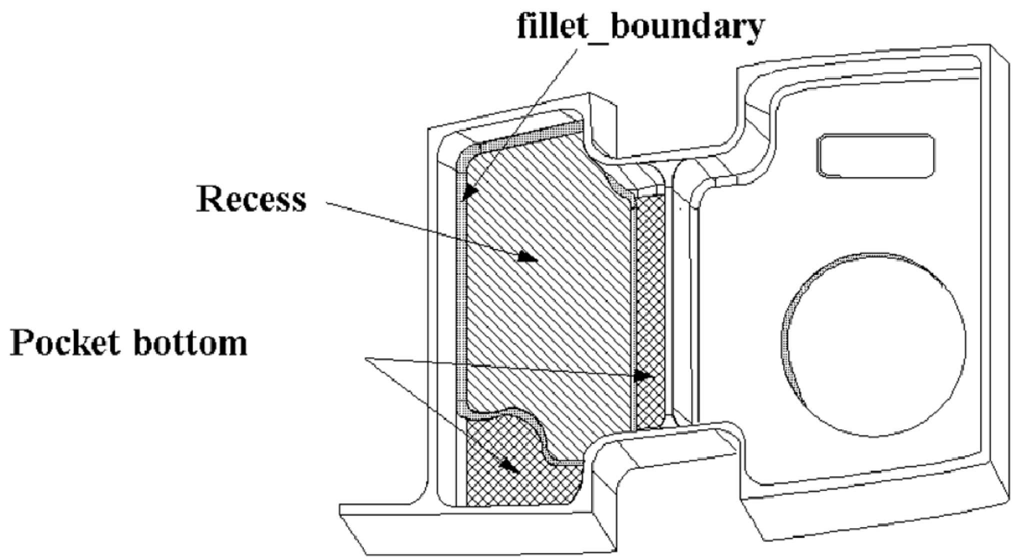


Figure 93 — Recess

4.2.193.2 fillet_boundary

The fillet_boundary specifies an outline or shape that is an enclosed area that shall be a close profile. The profile specifies the area required by a Recess. The placement of the fillet_boundary shall be with the origin of the Profile at the origin of the Recess. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Recess. See 4.3.241 for the application assertion.

4.2.193.3 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the recess. The Boss or Protrusion feature define the shape of the material that is to remain in the recess 4.3.243 or 4.3.244 for the application assertion.

4.2.194 Rectangular_boss

A Rectangular_boss is a type of Boss (see 4.2.11) that is an enclosed volume with opposite sides that are equal in length.

NOTE Figure 94 illustrates the Rectangular_boss. The lengths are determined in the profile view of the Boss.

The data associated with a Rectangular_boss are the following:

— change_in_boundary;

— rectangular_profile.

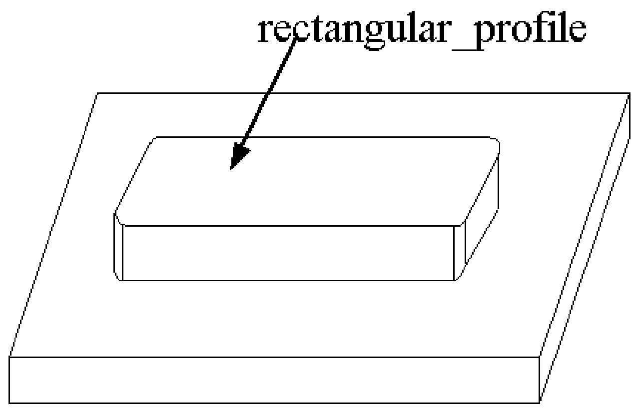


Figure 94 — Rectangular_boss

4.2.194.1 change_in_boundary

The change_in_boundary specifies a taper that defines the change in shape of the Rectangular_boss. The change_in_boundary need not be specified for a particular Rectangular_boss. See 4.3.245 or 4.3.246 for the application assertion.

4.2.194.2 rectangular_profile

The rectangular_profile specifies an enclosed area bounded by four sides with opposite sides equal in length and corners at 90 degrees. The orientation is at the center of the rectangle, the X-axis is parallel to the length of the rectangle and the Y-axis is parallel to the width. See 4.3.247 for the application assertion.

4.2.195 Rectangular_closed_pocket

A Rectangular_closed_pocket is a type of Pocket (see 4.2.177) that is an enclosed volume with opposite sides that are equal in length.

NOTE Figure 95 illustrates the Rectangular_closed_pocket.

The data associated with a Rectangular_closed_pocket are the following:

— closed_boundary;

— volume_not_removed.

4.2.195.1 closed_boundary

The closed_boundary specifies the outline or shape that is an enclosed area that has a completely closed profile. The profile specifies the area required by a Rectangular_closed_pocket. The placement of the closed_boundary shall be with the origin of the profile at the origin of the pocket. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Rectangular_closed_pocket. See 4.3.248 for the application assertion.

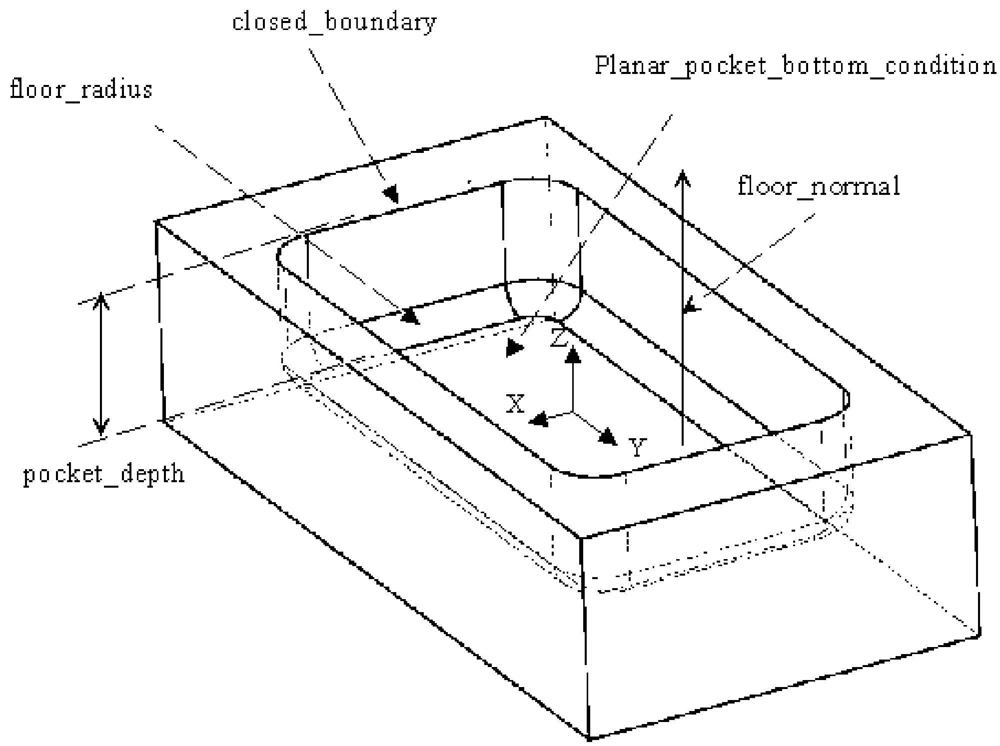


Figure 95 — Rectangular_closed_pocket

4.2.195.2 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss feature defines the shape of the material that is to remain in the pocket. See 4.3.249 or 4.3.250 for the application assertion.

4.2.196 Rectangular_closed_profile

A Rectangular_closed_profile is a type of Closed_profile (see 4.2.35) that is an enclosed area bounded by four sides with opposite sides equal in length and corners at 90 degrees. The orientation is at the center of the rectangle, the X-axis is parallel to the length of the rectangle and the Y-axis is parallel to the width.

NOTE Figure 96 illustrates the Rectangular_closed_profile.

The data associated with a Rectangular_closed_profile are the following:

— corner_radius;

— profile_length;

— profile_width.

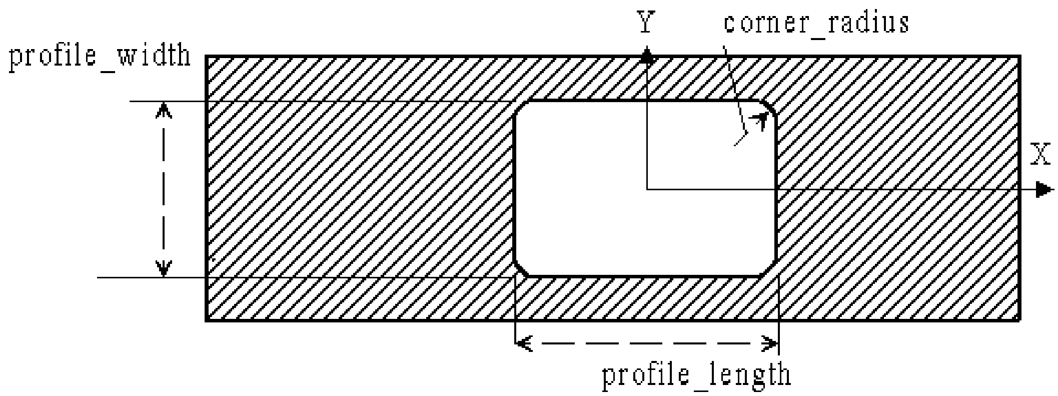


Figure 96 — Rectangular_closed_profile

4.2.196.1 corner_radius

The corner_radius specifies the size of the arc in all four corners of the rectangular profile. See 4.3.251 for the application assertion.

4.2.196.2 profile_length

The profile_length specifies the length of the side, along the X-axis, of the rectangular profile. See 4.3.251 for the application assertion.

4.2.196.3 profile_width

The profile_width specifies the length of the side, along the Y-axis, of the rectangular profile. See 4.3.251 for the application assertion.

4.2.197 Rectangular_closed_shape_profile

The Rectangular_closed_shape_profile is a type of Shape_profile (see 4.2.220) that is an enclosed volume with opposite sides that are equal in length. The data associated with a Rectangular_closed_shape_profile are the following:

— closed_boundary.

4.2.197.1 closed_boundary

The closed_boundary specifies the outline or shape that is an enclosed area that has a completely closed profile. The profile specifies the area required by a Rectangular_closed_shape_profile. The placement of the closed_boundary shall be with the origin of the profile at the origin of the profile. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Rectangular_closed_shape_profile. See 4.3.252 for the application assertion.

4.2.198 Rectangular_offset_pattern

A Rectangular_offset_pattern is a modification of the placement of a particular occurrence of the base feature in a Rectangular_pattern relative to its expected placement.

NOTE Figure 97 illustrates the Rectangular_offset_pattern.

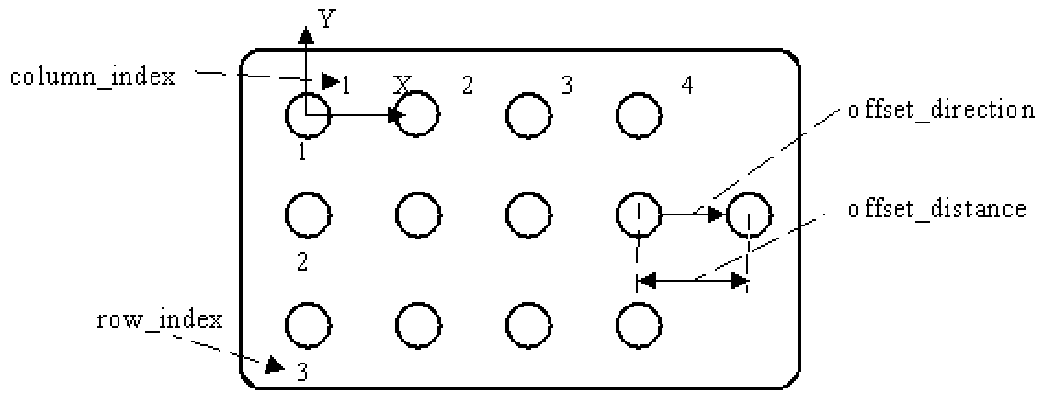


Figure 97 — Rectangular_offset_pattern

The data associated with a Rectangular_offset_pattern are the following:

— column_index;

— offset_direction;

— offset_distance;

— row_index.

4.2.198.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.254 for the application assertion.

4.2.198.2 offset_direction

The offset_direction specifies the direction to offset a base feature from its original position in the rectangular pattern. See 4.3.253 for the application assertion.

4.2.198.3 offset_distance

The offset_distance specifies the amount of offset from a feature location in a Rectangular_pattern for placing another feature. See 4.3.254 for the application assertion.

4.2.198.4 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.254 for the application assertion.

4.2.199 Rectangular_omit_pattern

A Rectangular_omit_pattern is an omission of a particular occurrence of the base feature in an Rectangular_pattern.

NOTE Figure 98 illustrates Rectangular_omit_pattern.

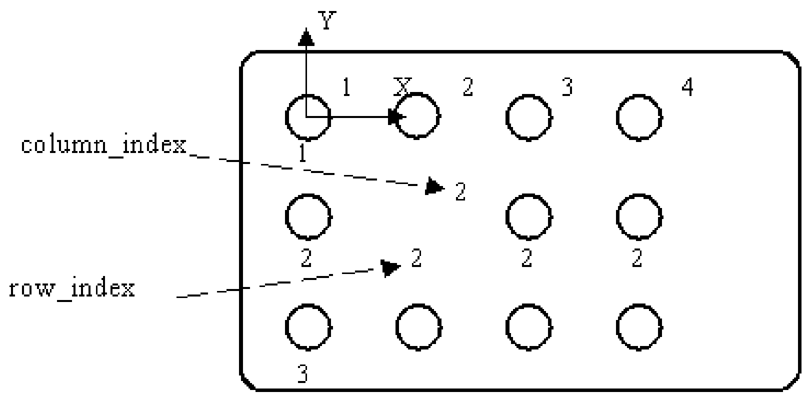


Figure 98 — Rectangular_omit_pattern

The data associated with a Rectangular_omit_pattern are the following:

— column_index;

— row_index.

4.2.199.1 column_index

The column_index specifies the unique identification for a feature in a column of multiple features. See 4.3.255 for the application assertion.

4.2.199.2 row_index

The row_index specifies the unique identification for a feature in a row of multiple features. See 4.3.255 for the application assertion.

4.2.200 Rectangular_open_pocket

A Rectangular_open_pocket is a type of Pocket (see 4.2.177) that is an open profile with opposite sides that are of equal length and with one side that does not make contact with the part. The open boundary is defined by a Square_U_profile (see 4.2.231) such that, when swept along a path, the profile defines the area on a part for volume removal. The orientation and placement of the Square_U_profile shall be the same as the Rectangular_open_pocket.

NOTE Figure 99 illustrates the Rectangular_open_pocket.

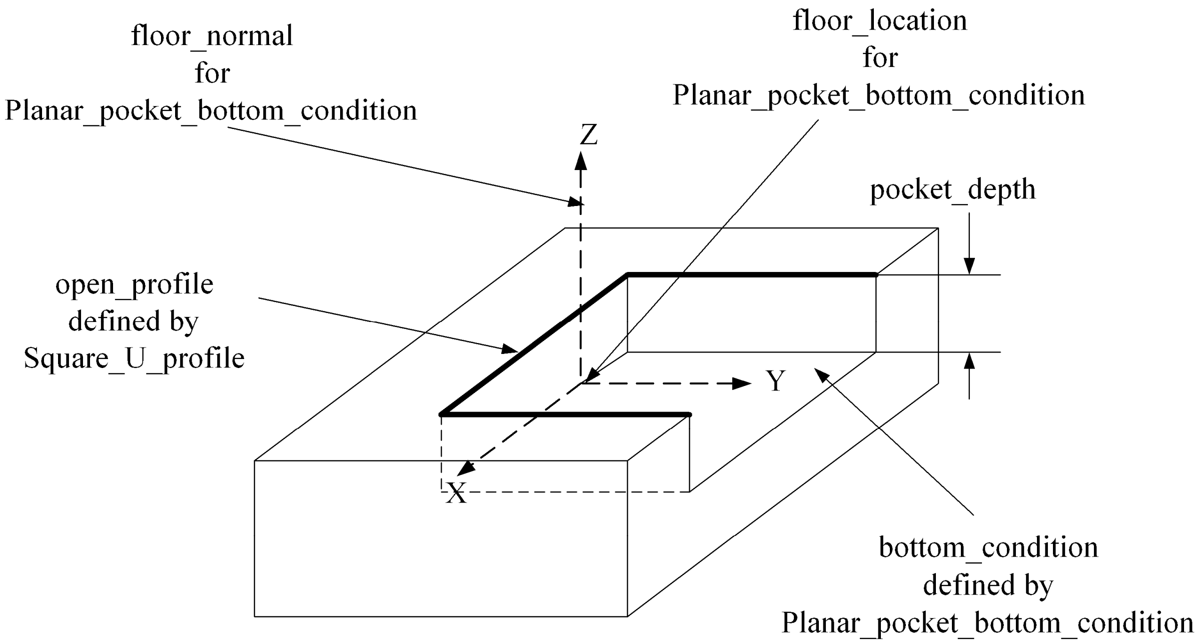


Figure 99 — Rectangular_open_pocket

The data associated with a Rectangular_open_pocket are the following:

— open_boundary;

— volume_not_removed.

4.2.200.1 open_boundary

The open_boundary specifies the outline or shape that is an enclosed area that is open on one side. The profile specifies the area required by a Rectangular_open_pocket. The placement of the open boundary shall be with the origin of the profile at the origin of the pocket. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Rectangular_open_pocket. See 4.3.258 for the application assertion.

4.2.200.2 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the pocket. The Boss feature defines the shape of the material that is to remain in the pocket. See 4.3.256 or 4.3.257 for the application assertion.

4.2.201 Rectangular_open_shape_profile

The Rectangular_open_shape_profile is a type of Shape_profile (see 4.2.220) that is an open profile with opposite sides that are of equal length and with one side that does not make contact with the part. The data associated with a Rectangular_open_shape_profile are the following:

— open_boundary.

4.2.201.1 open_boundary

The open_boundary specifies the outline or shape that is an enclosed area that is open on one side. The profile specifies the area required by a Rectangular_open_shape_profile. The placement of the open_boundary shall be with the origin of the profile at the origin of the feature. The X-axis and Y-axis of the profile shall be the same as the X-axis and Y-axis of the Rectangular_open_shape_profile. See 4.3.259 for the application assertion.

4.2.202 Rectangular_pattern

A Rectangular_pattern is a type of Replicate_feature (see 4.2.204) that is a shape component arranged in a pattern of rows and columns.

NOTE Figure 100 illustrates the Rectangular_pattern.

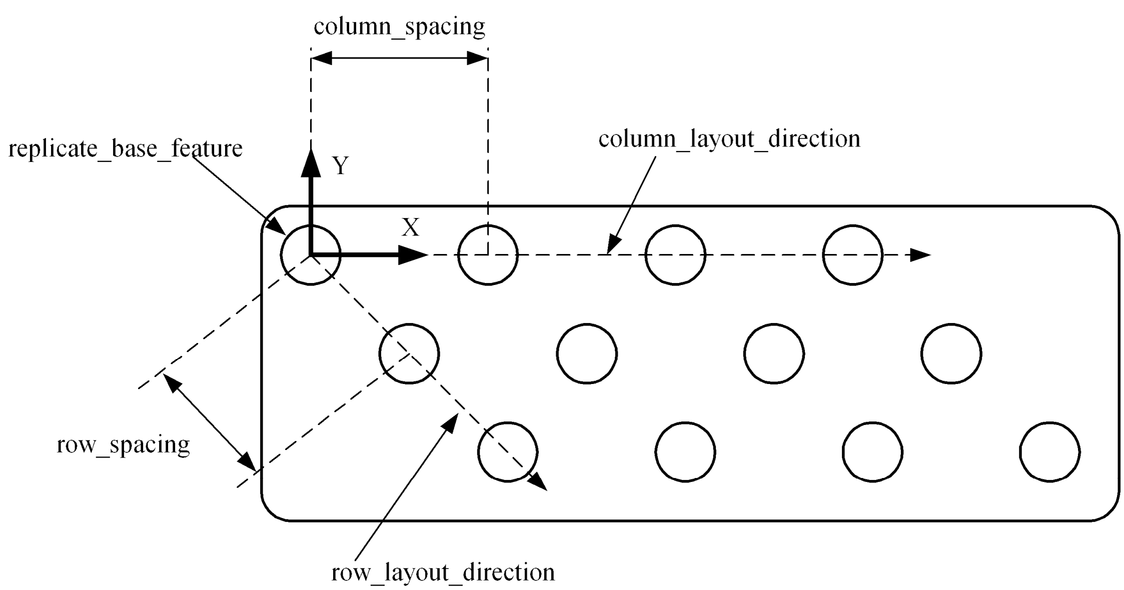


Figure 100 — Rectangular_pattern

The data associated with a Rectangular_pattern are the following:

— column_layout_direction;

— column_spacing;

— columns;

— missing_base_feature;

— relocated_base_feature;

— row_layout_direction;

— row_spacing;

— rows.

4.2.202.1 column_layout_direction

The column_layout_direction specifies the linear direction for defining the columns of the pattern. See 4.3.260 for the application assertion.

4.2.202.2 column_spacing

The column_spacing specifies the amount of space between features in a Rectangular_pattern column. See 4.3.261 for the application assertion.

4.2.202.3 columns

The columns specifies the number of columns for placing features in the Rectangular_pattern. See 4.3.261 for the application assertion.

4.2.202.4 missing_base_feature

The missing_base_feature specifies the definition to remove any number of base features from the Rectangular_pattern. The missing_base_feature need not be specified for a particular Rectangular_-pattern. There may be more than one missing_base_feature for a Rectangular_pattern. See 4.3.263 for the application assertion.

4.2.202.5 relocated_base_feature

The relocated_base_feature specifies the definition to offset any number of base features from the Rectangular_pattern. The relocated_base_feature need not be specified for a particular Rectangular_-pattern. There may be more than one relocated_base_feature for a Rectangular_pattern. See 4.3.262 for the application assertion.

4.2.202.6 row_layout_direction

The row_layout_direction specifies the linear direction for defining the rows of the pattern. See 4.3.260 for the application assertion.

4.2.202.7 row_spacing

The row_spacing specifies the amount of space between features in a Rectangular_pattern row. See 4.3.261 for the application assertion.

4.2.202.8 rows

The rows specifies the number of rows for placing features in the Rectangular_pattern. See 4.3.261 for the application assertion.

4.2.203 Replicate_base

A Replicate_base is the type of feature to be used as a base feature for reproduction. The Replicate_base shall have a base defined by either a Machining_feature (see 4.2.126) or a Replicate_feature (see 4.2.204). The data associated with a Replicate_base are the following:

— base_feature.

4.2.203.1 base_feature

The base_feature specifies the feature that will be reproduced by the Replicate_feature. The base_feature may be either a Machining_feature (see 4.2.126) or a Replicate_feature (see 4.2.204). See 4.3.264 and 4.3.265 for the application assertion.

4.2.204 Replicate_feature

A Replicate_feature is a type of Machining_feature (see 4.2.126) that is a basis shape, and the arrangement of identical copies of that base shape. Each base shape is a Machining_feature oriented to the first defined position of a pattern. The patterns describe how to replicate that feature to different placements on the part. Each Replicate_feature is either a Circular_pattern (see 4.2.31), a General_pattern (see 4.2.95), or a Rectangular_pattern (see 4.2.202). The data associated with a Replicate_feature are the following:

— placement;

— replicate_base_feature.

4.2.204.1 placement

The placement specifies the position and orientation of a Replicate_feature relative to the base shape for a part. See 4.3.266 for the application assertion.

4.2.204.2 replicate_base_feature

The replicate_base_feature specifies the feature that will be replicated by the Replicate_feature. See 4.3.267 for the application assertion.

4.2.205 Requisition

A Requisition is a document that contains an order for equipment or materials to perform or support the manufacturing process. Each Requisition may be one of the following: a Cutting_tool_requisition (see 4.2.51), a Dedicated_fixture_requisition (see 4.2.58), a Machine_requisition (see 4.2.125), a Material_-requisition (see 4.2.135), a Modular_fixture_requisition (see 4.2.139), or an Indirect_stock_requisition (see 4.2.113). The data associated with a Requisition are the following:

— quantity_ordered;

— required_delivery_date;

— requisition_date;

— requisition_description;

— requisition_number.

4.2.205.1 quantity_ordered

The quantity_ordered specifies the number of items that are being purchased by an organization using the Requisition.

4.2.205.2 required_delivery_date

The required_delivery_date specifies the year, month and day on which the item or items specified in the Requisition are required at the site that ordered them.

4.2.205.3 requisition_date

The requisition_date specifies the year, month, and day on which the Requisition was issued.

4.2.205.4 requisition_description

The requisition_description specifies human interpretable prose that describes the item being ordered, and provides any special instructions that may pertain to the order.

4.2.205.5 requisition_number

The requisition_number specifies a unique identification of the Requisition.

4.2.206 Resource_acquisition_order

A Resource_acquisition_order is a document required for process planning of a part. This order defines the requirements for material needed to produce a part.

EXAMPLE The Resource_acquisition_order may contain customer order information, part identification, quantity, or schedules.

The data associated with a Resource_acquisition_order are the following:

— order_id.

4.2.206.1 order_id

The order_id specifies a unique identifier for the Resource_acquisition_order.

4.2.207 Revolved_feature

A Revolved_feature is a type of Machining_feature (see 4.2.126) that is a sweeping of a planar shape one complete revolution about an axis. The planar shape shall be finite in length, coplanar with the axis of revolution, and shall not intersect the axis of revolution. The axis of revolution shall be the same as the Z-axis of the feature. The Revolved_feature may be either an outer shape of a part or a volume removal, depending on the material direction. Each Revolved_feature is either a General_revolution (see 4.2.100), a Groove (see 4.2.106), a Revolved_flat (see 4.2.208), or a Revolved_round (see 4.2.209).

The data associated with a Revolved_feature are the following:

— material_side;

— radius.

4.2.207.1 material_side

The material_side specifies the material direction. The direction of removal indicates the direction the material will be removed from the part. See 4.3.269 for the application assertion.

4.2.207.2 radius

The radius specifies the distance from the axis of rotation to define placement of the profile that will be swept about the axis. See 4.3.268 for the application assertion.

4.2.208 Revolved_flat

A Revolved_flat is a type of Revolved_feature (see 4.2.207) that is the sweeping of a straight line about an axis.

NOTE Figure 101 illustrates the Revolved_flat.

The data associated with a Revolved_flat are the following:

— flat_edge_shape.

4.2.208.1 flat_edge_shape

The flat_edge_shape specifies the line with direction and magnitude that when revolved about an axis defines the area on a part for volume removal. The placement of the profile shall be along the X-axis of the Revolved_flat at a specified distance away from the origin. The Y-axis orientation of the Linear_profile shall be the same as the Y-axis of the Revolved_flat, the X-axis and Z-axis are independent of the orientation of the Revolved_flat feature. See 4.3.270 for the application assertion.

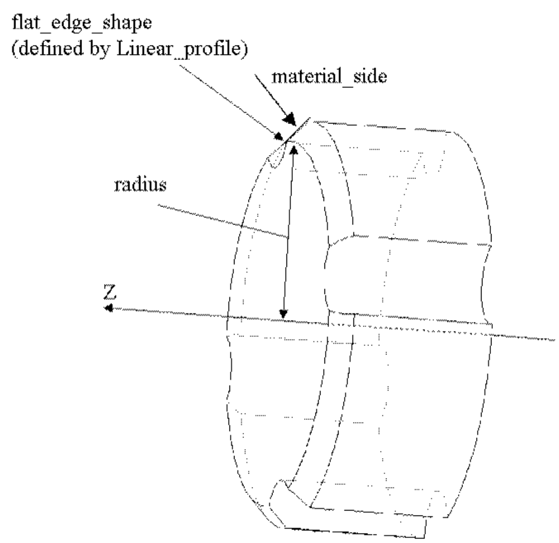


Figure 101 — Revolved_flat

4.2.209 Revolved_round

A Revolved_round is a type of Revolved_feature (see 4.2.207) that is the sweeping of an arc about an axis.

NOTE Figure 102 illustrates the Revolved_round.

The data associated with a Revolved_round are the following:

— rounded_edge_shape.

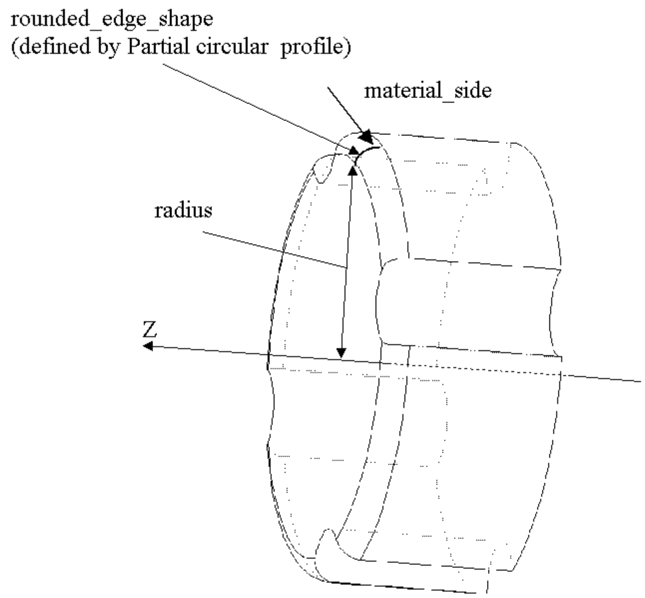


Figure 102 — Revolved_round

4.2.209.1 rounded_edge_shape

The rounded_edge_shape specifies the arc that when revolved about an axis defines the area on a part for volume removal. The placement of the profile shall be along the X-axis of the Revolved_round at a specified distance away from the origin. The Z-axis orientation of the Partial_circular_profile shall be the same as the Y-axis of the Revolved_round, the X-axis and Y-axis are independent of the orientation of the Revolved_round feature. See 4.3.271 for the application assertion.

4.2.210 Rib_top

The Rib_top is a type of Multi_axis_feature (see 4.2.140) that is the removal of a volume to a floor with no sides, or floor radius. Rib_top features may adjoin with another Rib_top feature.

EXAMPLE The material that separates two pockets on a part is an example of a rib. The top surface of that rib would be an example of a Rib_top.

NOTE Figure 103 illustrates the Rib_top feature.

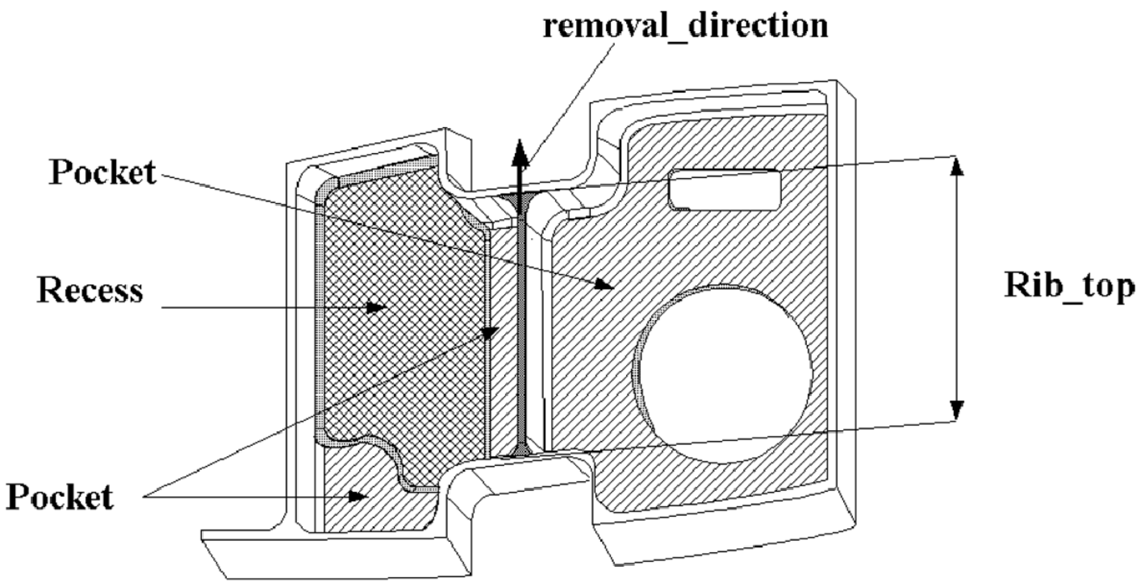


Figure 103 — Rib_top

The data associated with a Rib_top are the following:

— floor_condition;

— removal_direction.

4.2.210.1 floor_condition

A floor_condition specifies the bottom state of a Rib_top. The floor may be flat or any arbitrary shape. See 4.3.273 for the application assertion.

4.2.210.2 removal_direction

A removal_direction specifies a vector that points in the general direction away from the material for a Rib_top. See 4.3.272 for the application assertion.

4.2.211 Rib_top_floor

A Rib_top_floor is the bottom restriction for a Rib_top (see 4.2.210). The bottom may be flat, or any arbitrary shape. Each Rib_top_floor is either a General_rib_top_floor (see 4.2.101), or a Planar_rib_-top_floor (see 4.2.174).

4.2.212 Round_hole

A Round_hole is a type of Hole (see 4.2.111) that is a removal of a volume of cylindrical shape from a part. A Round_hole need not be tapered. The orientation is at a point in the bottom of the hole. The Z-axis is along the centerline with the direction out of the hole.

NOTE Figure 104 illustrates the Round_hole with and without a taper.

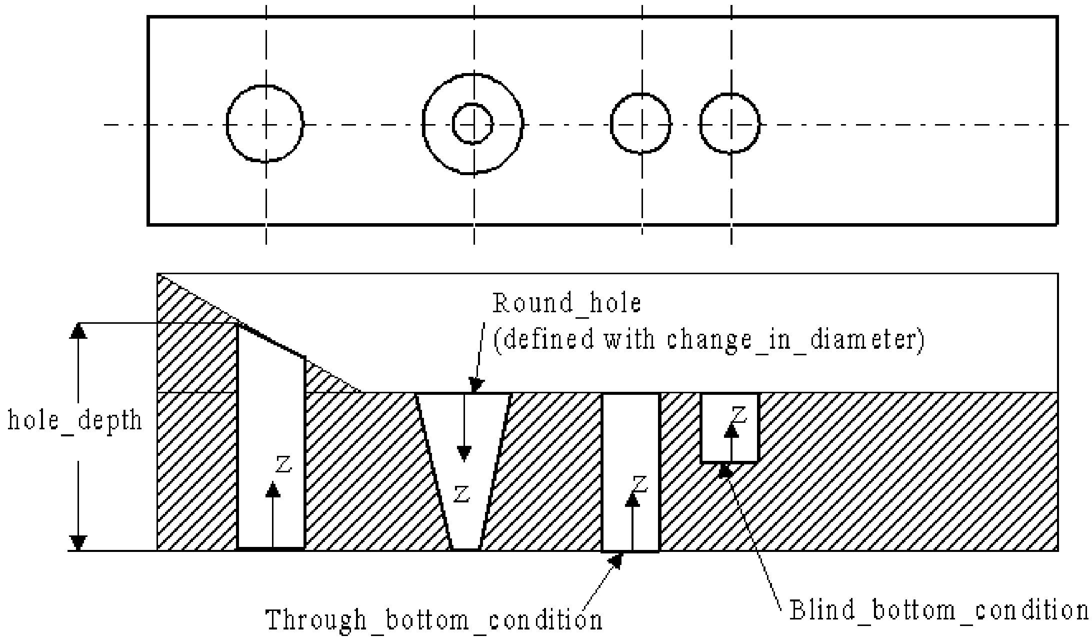


Figure 104 — Round_hole

The data associated with a Round_hole are the following:

— bottom_condition;

— change_in_diameter;

— diameter;

— hole_depth.

4.2.212.1 bottom_condition

The bottom_condition specifies the shape of the bottom of a Round_hole feature. Each bottom_condition may be one of the following: a Blind_bottom_condition (see 4.2.9), or a Through_bottom_condition (see 4.2.249). See 4.3.274 and 4.3.275 for the application assertion.

4.2.212.2 change_in_diameter

The change_in_diameter specifies the taper that defines the change in shape of the Round_hole. The change_in_diameter need not be specified for a particular Round_hole. See 4.3.278, 4.3.279 and 4.3.280 for the application assertion.

4.2.212.3 diameter

The diameter is the distance across a Round_hole. The placement and orientation of the Circular_-closed_profile shall be the same as the Round_hole feature. See 4.3.276 for the application assertion.

4.2.212.4 hole_depth

The hole depth is some amount of distance from the bottom of a Round_hole to a point that is outside of the Round_hole feature. Hole depth places a limitation on the Round_hole depth definition so not to interfere with other features that might be nearby. The placement and orientation of the Linear_path shall be the same as the Round_hole feature. See 4.3.277 for the application assertion.

4.2.213 Rounded_end

A Rounded_end is a type of Multi_axis_feature(see 4.2.140) that is a partially circular shape passed along a linear path.

NOTE Figure 105 illustrates the Rounded_end.

The data associated with a Rounded_end are the following:

— course_of_travel;

— partial_circular_boundary.

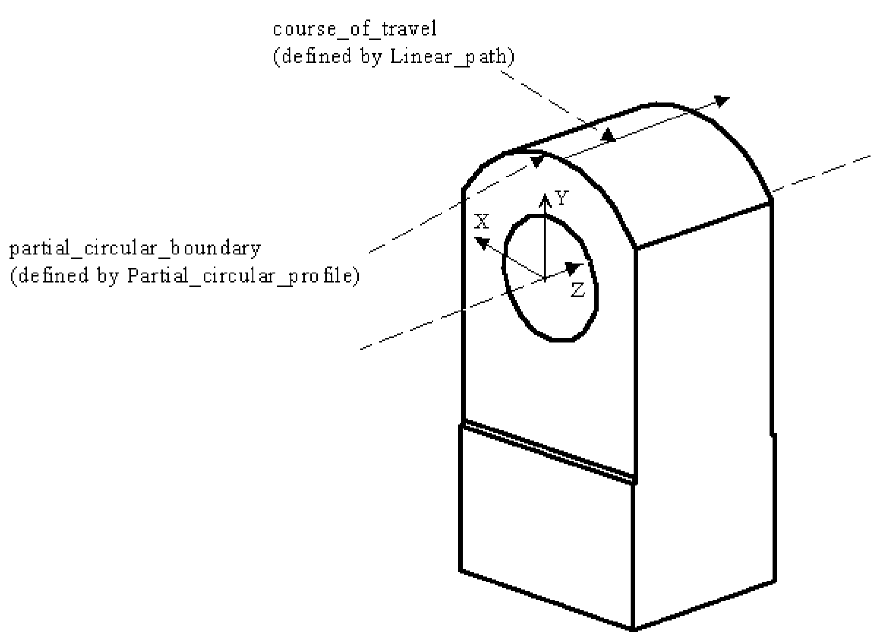


Figure 105 — Rounded_end

4.2.213.1 course_of_travel

The course_of_travel specifies a straight line with magnitude and direction. The placement and orientation of the Linear_path that defines the course_of_travel shall be the same as the Rounded_end feature. See 4.3.281 for the application assertion.

4.2.213.2 partial_circular_boundary

The partial_circular_boundary specifies the arc that when swept along a path defines the area on a part for volume removal. The placement and orientation of the Partial_circular_profile that defines the partial_circular_boundary shall be the same as the Rounded_end feature. See 4.3.282 for the application assertion.

4.2.214 Rounded_U_profile

A Rounded_U_profile is a type of Open_profile (see 4.2.145) that is a shape bounded by two parallel lines and a semicircle. Each line begins at the end point of the semicircle. The lines are tangent to the circle and extend infinitely. The profile is positioned with the opening in the direction of the Y-axis. The orientation is at a point on the profile the farthest distance from the opening measured along the Y-axis. The X-axis is tangent to the semicircle.

NOTE Figure 106 illustrates the Rounded_end.

The data associated with a Rounded_U_profile are the following:

— depth;

— width.

4.2.214.1 width

The width specifies the distance across a Round_U_profile. See 4.3.283 for the application assertion.

NOTE Figure 106 illustrates the Rounded_U_profile.

4.2.214.2 depth

The depth specifies the distance, measured along the Y axis, from the profile origin to the top of the profile. See 4.3.283 for the application assertion.

4.2.215 Second_chamfer_offset

A Second_chamfer_offset is a choice of methods for creating a Chamfer feature. A Chamfer requires an offset and the choice between a second offset or an angle. Each Second_chamfer_offset is either a Chamfer_angle (see 4.2.23) or a Second_offset (see 4.2.216). The data associated with a Second_-chamfer_offset are the following:

— second_face.

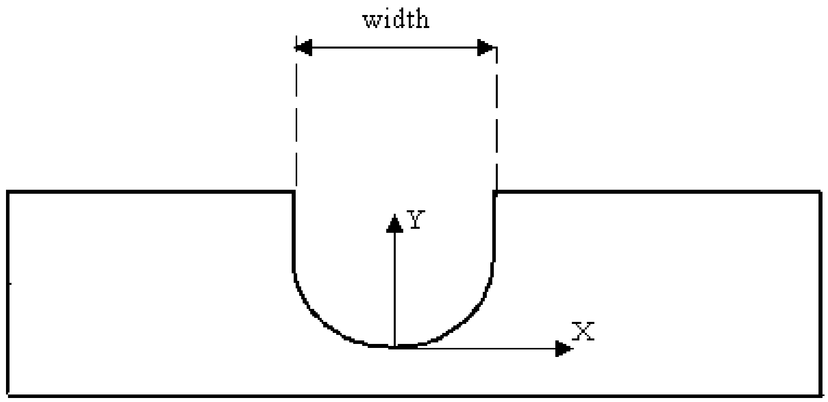


Figure 106 — Rounded_U_profile

4.2.215.1 second_face

The second_face is one of two faces the Chamfer feature will transition between. The first face is specified by First_offset (see 4.2.82). See 4.3.284 for the application assertion.

4.2.216 Second_offset

A Second_offset is a type of Second_chamfer_offset (see 4.2.215) that is the amount of length offset from a face for creating a Chamfer feature.

NOTE Figure 15 illustrates a Second_offset for a Chamfer feature.

The data associated with a Second_offset are the following:

— offset_amount.

4.2.216.1 offset_amount

The offset_amount specifies the offset value from the edge of a face to the Chamfer face. See 4.3.285 for the application assertion.

4.2.217 Shape

A Shape is the physical form of the part that is being machined. The data associated with a Shape are the following:

— base_shape_definition;

— B-rep_form;

— element.

4.2.217.1 base_shape_definition

The base_shape_definition specifies either the implicit or the explicit definition of the Part. See 4.3.286 for the application assertion.

4.2.217.2 B-rep_form

The B-rep_form specifies the boundary representation shape of the Part. The B-rep_form need not be specified for a particular Shape. There may be more than one B-rep_form for a Shape. See 4.3.287 for the application assertion.

4.2.217.3 element

The element specifies the components of the shape of the Part. The element need not be specified for a particular Shape. There may be more than one element for a Shape. See 4.3.288 for the application assertion.

4.2.218 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:

— B-rep_form;

— B-rep_shape;

— element.

4.2.218.1 B-rep_form

The B-rep_form specifies aspects of the boundary representation of the Part. The B-rep_form need not be specified for a particular Shape. There may be more than one B-rep_form for a Shape_aspect. There may be more than one B-rep_form for a Shape_aspect. See 4.3.290 for the application assertion.

4.2.218.2 B-rep_shape

The B-rep_shape specifies the boundary representation of the shape of the Part. The B-rep_shape need not be specified for a particular Shape_aspect. There may be more than one B-rep_shape for a Shape_-aspect. See 4.3.289 for the application assertion.

4.2.218.3 element

The element specifies components of the shape of the Part. The element need not be specified for a particular Shape_aspect. See 4.3.291 for the application assertion.

4.2.219 Shape_element

A Shape_element is a specific kind of Shape_aspect that identifies a portion of a shape and its representation. Each Shape_element is either a Direction_element (see 4.2.71), a Face_shape_element (see 4.2.79), a Location_element (see 4.2.122), a Manufacturing_feature (see 4.2.129), a Path_element (see 4.2.164) or a Planar_element (see 4.2.170).

4.2.220 Shape_profile

A Shape_profile is a type of Profile_feature (see 4.2.182) that is the removal volume of raw stock or other excess material from the boundary shape of a part. The sides of a profile may be parallel to the profile's orientation vector, or may be parallel with an offset amount. The placement may be at the bottom of the profile with the Z-axis in the direction toward the top of the profile, or at the top of the profile with the Z-axis in the direction toward the bottom of the profile. The bottom of the boundary shape is limited by a floor condition. Each Shape_profile is either a Circular_closed_shape_profile (see 4.2.26), a Partial_-circular_shape_profile (see 4.2.162), a General_shape_profile (see 4.2.102), a Rectangular_closed_-shape_profile (see 4.2.197), or a Rectangular_open_shape_profile (see 4.2.201).

NOTE Figure 107 illustrates a Shape_profile.

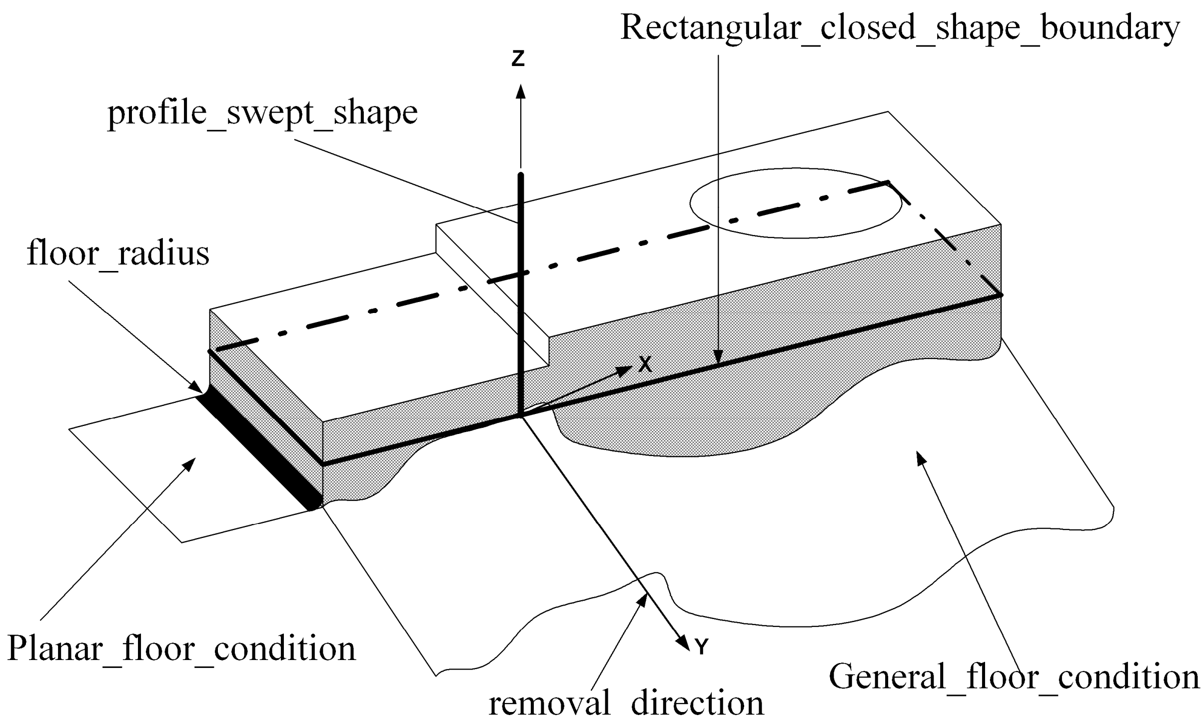


Figure 107 — Shape_profile

The data associated with a Shape_profile are the following:

— floor_condition;

— removal_direction.

4.2.220.1 floor_condition

The floor_condition specifies the shape of the bottom of a Shape_profile feature. The floor_condition is either a Profile_floor (see 4.2.183) or a Through_profile_floor (see 4.2.251). See 4.3.293 and 4.3.294 for the application assertions.

4.2.220.2 removal_direction

A removal_direction specifies a vector that points in the general direction away from the material for a Shape_profile. See 4.3.292 for the application assertion.

4.2.221 Shop_work_order

A Shop_work_order is a document that contains the information to process a part on the shop floor, and track the part's progress throughout its manufacture. The data associated with a Shop_work_order are the following:

— order_id.

4.2.221.1 order_id

The order_id specifies a unique identifier for the Shop_work_order.

4.2.222 Single_piece_part

A Single_piece_part is a type of Part (see 4.2.155) that is the physical item which is intended to be produced through the manufacturing process. The Single_piece_part may be an Manufactured_assembly (see 4.2.135). The data associated with a Single_piece_part are the following:

— alternate_material_definition;

— material_definition.

4.2.222.1 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. See 4.3.295 for the application assertion.

4.2.222.2 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 4.3.296 for the application assertion.

4.2.223 Size_tolerance

A Size_tolerance is a type of Dimensional_tolerance (see 4.2.69) that is the size dimension tolerance characteristic for a geometric element. Each Size_tolerance is either an Angular_size_dimension_-tolerance (see 4.2.4), a Curved_dimension_tolerance (see 4.2.48), a Diameter_dimension_tolerance (see 4.2.65), a Height_dimension (see 4.2.108), a Length_dimension (see 4.2.115), a Width_dimension (see 4.2.261), an Externally_defined_size_dimension (see 4.2.77), a Thickness_tolerance (see 4.2.248) or a Radial_dimension_tolerance (see 4.2.191). The data associated with a Size_tolerance are the following:

— applied_shape;

— envelope.

4.2.223.1 applied_shape

The applied_shape specifies the physical shape of the Part that is being toleranced. See 4.3.297 for the application assertion.

4.2.223.2 envelope

The envelope specifies that each geometric constraint has to be fulfilled in itself. The envelope of the perfect shape corresponding to the maximum material shall not be larger than the specified dimension and tolerance. The envelope attribute shall be a boolean value, if TRUE the envelope is required for the Size_tolerance.

4.2.224 Slot

A Slot is a type of Multi_axis_feature(see 4.2.140) that is a channel or depression with continuous direction of travel. The Slot origin shall be located at one end of the slot, the Z-axis shall indicate the direction of the slot, and the Y-axis shall indicate the direction away from the part. Slot is either a Closed_slot (see 4.2.36) or an Open_slot (see 4.2.146).

NOTE Figure 108 illustrates types of Slots.

The data associated with a Slot are the following:

— sweep_shape.

4.2.224.1 sweep_shape

The sweep_shape specifies the implicit 2D profile definition that, when combine with a Path, creates the shape of the Slot. See 4.3.298 for the application assertion.

4.2.225 Slot_end_type

A Slot_end_type is the end conditions of a slot. A slot shall have two ends, each end shall be open or closed. Each Slot_end_type is either a Flat_slot_end_type (see 4.2.84), a Open_slot_end_type (see 4.2.147), a Radiused_slot_end_type (see 4.2.192), or a Woodruff_slot_end_type (see 4.2.262).

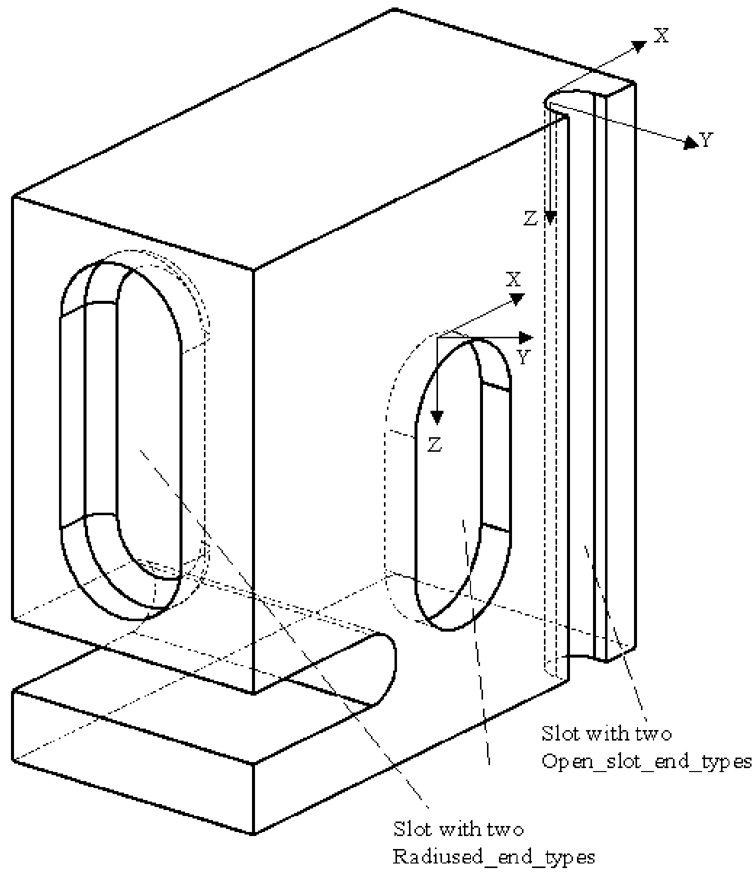


Figure 108 — Slot

The data associated with a Slot_end_type are the following:

— first_or_second.

4.2.225.1 first_or_second

The first_or_second specifies a value of 'FIRST' if the Slot_end_type is closest to the positioning point of a Slot or 'SECOND' if it is the farthest away.

4.2.226 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:

— constraint;

— release_date;

— specification_class;

— specification_description;

— specification_id;

— version_id.

4.2.226.1 constraint

The constraint specifies the restriction on the Specification. The constraint need not be specified for a particular Specification. There may be more than one constraint for a Specification. See 4.3.299 for the application assertion.

4.2.226.2 release_date

The release_date specifies the date of the applicable version of the specification. The release_date need not be specified.

4.2.226.3 specification_class

The specification_class specifies a section within a Specification that is divided into classes. A Specification may but need not require a specification_class.

4.2.226.4 specification_description

The specification_description specifies in human interpretable prose a description of the contents of the specification and any notes with respect to the Specification. A Specification may but need not require a specification_description.

4.2.226.5 specification_id

The specification_id specifies a unique identifier of the document.

4.2.226.6 version_id

The version_id specifies the revision or issue number that uniquely distinguishes one Specification object from other versions of Specification objects with the same specification_id. The version_id need not be specified.

4.2.227 Specification_usage_constraint

A Specification_usage_constraint is a restriction on the application of information defined within a Specification. The data associated with a Specification_usage_constraint are the following:

— class_id;

— element.

4.2.227.1 class_id

The class_id specifies the data or range of data with respect to the element that defines the restriction imposed on the usage of the Specification.

4.2.227.2 element

The element specifies the particular piece or area of information that is being restricted within the Specification.

4.2.228 Spherical_cap

A Spherical_cap is a type of Machining_feature (see 4.2.126) that is circular about an axis of rotation. A Spherical_cap consists of all points a given distance from a point constituting its center. The Z-axis shall be in the direction away from the material.

NOTE Figure 109 illustrates the Spherical_cap.

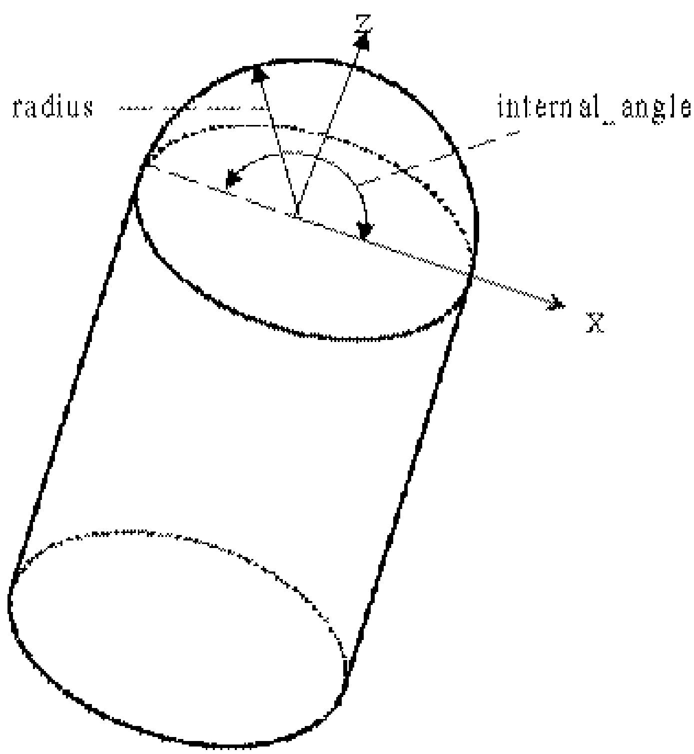


Figure 109 — Spherical_cap

The data associated with a Spherical_cap are the following:

— internal_angle;

— radius.

4.2.228.1 internal_angle

The internal_angle specifies the size of an angle from an axis for defining a portion of a sphere to use as a spherical_cap feature. The X-axis defines the start of the spherical_cap and the internal_angle is measured from this axis. See 4.3.300 for the application assertion.

4.2.228.2 radius

The radius specifies the constant distance from a point for defining a sphere. See 4.3.300 for the application assertion.

4.2.229 Spherical_hole_bottom

A Spherical_hole_bottom is a type of Blind_bottom_condition (see 4.2.9) that is a bottom of a Round_-hole which is concentric about an axis and defined by a radius. The radius is the same as the radius of the hole.

NOTE Figure 110 illustrates the Spherical_hole_bottom.

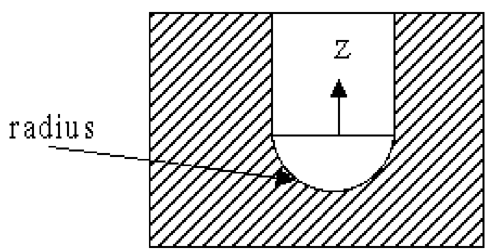


Figure 110 — Spherical_hole_bottom

The data associated with Spherical_hole_bottom are the following:

— radius.

4.2.229.1 radius

The radius specifies the radius at the bottom of the Round_hole. See 4.3.301 for the application assertion.

4.2.230 Spur_gear

A Spur_gear is a type of Defined_gear (see 4.2.59) that is a cylindrical gear whose tooth traces are straight line generators of the reference cylinder.

NOTE Figure 111 illustrates the Spur_gear.

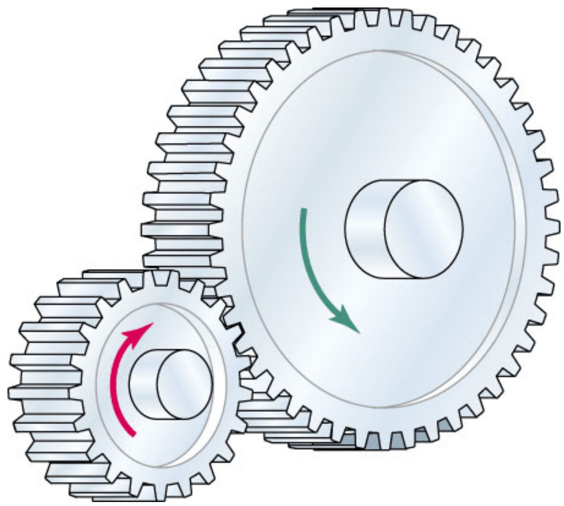


Figure 111 — Spur_gear

4.2.231 Square_U_profile

A Square_U_profile is a type of Open_profile (see 4.2.145) that is a shape bounded by three lines. One is the base line and has a defined length. The other two lines begin at the ends of the base line, and extend infinitely at any angle to the base line greater than zero degrees and less than 180 degrees. The two lines may also be at right angle to the base line. The corners of the Square_U_profile need not be blended by a radius.

NOTE Figure 112 illustrates the Square_U_profile.

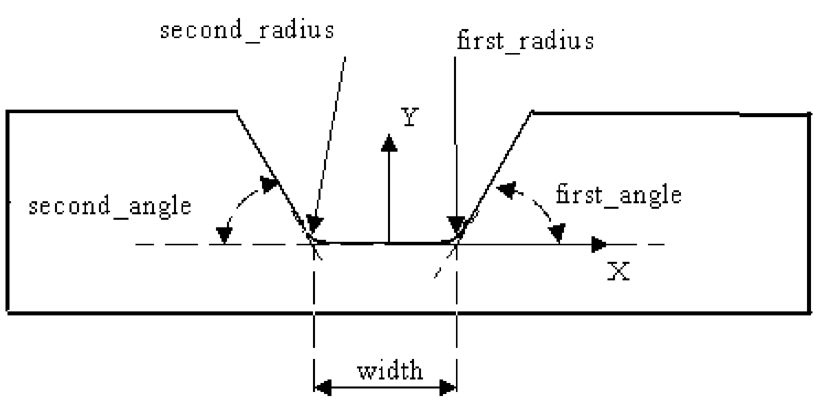


Figure 112 — Square_U_profile

The data associated with Square_U_profile are the following:

— depth;

— first_angle;

— first_radius;

— second_angle;

— second_radius;

— width.

4.2.231.1 depth

The depth specifies the distance, measured along the Y axis, from the profile origin to the top of the profile. See 4.3.302 for the application assertion.

4.2.231.2 first_angle

The first_angle specifies the size of an angle between one side of the profile and the base. See 4.3.302 for the application assertion.

4.2.231.3 first_radius

The first_radius specifies the radius shape blend between one side of the profile and the base. See 4.3.302 for the application assertion.

4.2.231.4 second_angle

The second_angle specifies the size of an angle between the second side of the profile and the base. See 4.3.302 for the application assertion.

4.2.231.5 second_radius

The second_radius specifies the radius shape blend between the second side of the profile and the base. See 4.3.302 for the application assertion.

4.2.231.6 width

The width specifies the size of the base line for a Square_U_profile. See 4.3.302 for the application assertion.

4.2.232 Step

A Step is a type of Multi_axis_feature(see 4.2.140) that is a linear sweep of a shape. The shape shall be specified by two lines that connect at a point and extend infinitely. The enclosed angle shall be smaller than 180 degrees. The intersection of the two lines need not be blended with a radius.

NOTE Figure 113 illustrates the Step.

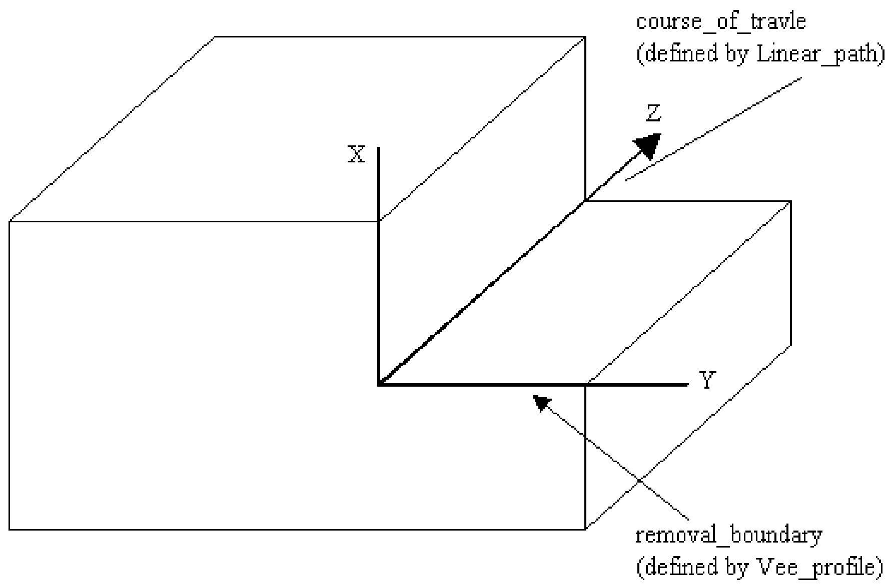


Figure 113 — Step

The data associated with a Step are the following:

— course_of_travel;

— removal_boundary;

— volume_not_removed.

4.2.232.1 course_of_travel

The course_of_travel specifies the straight line with magnitude and direction. The placement and orientation of the Linear_path shall be the same as the Step feature. See 4.3.304 for the application assertion.

4.2.232.2 removal_boundary

The removal_boundary specifies the a Vee_profile (see 4.2.260) that when swept along a path defines the area on a part for volume removal. The placement and orientation of the Vee_profile shall be the same as the Step feature. See 4.3.303 for the application assertion.

4.2.232.3 volume_not_removed

The volume_not_removed specifies an amount of material that is not to be removed from the step. The Boss or Protrusion feature define the shape of the material that is to remain in the pocket. See 4.3.305 or 4.3.306 for the application assertion.

4.2.233 Straight_bevel_gear

The Straight_bevel_gear is a type of Bevel_gear(see 4.2.8) that have straight tooth elements, which if extended, would pass through the point of intersection of their axes.

4.2.234 Straight_knurl

A Straight_knurl is a type of Turned_knurl (see 4.2.259) that is typified by knurl scoring that is parallel to the axis of the scored surface.

NOTE Figure 114 illustrates the Straight_knurl.

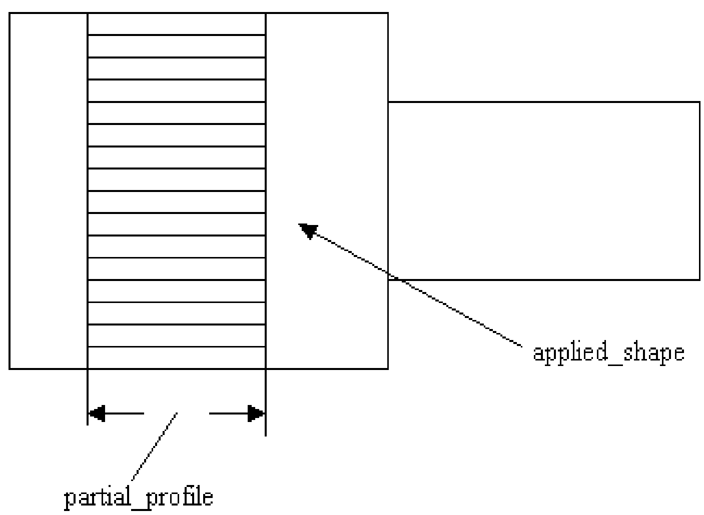


Figure 114 — Straight_knurl

4.2.235 Straightness_tolerance

A Straightness_tolerance is a type of Geometric_tolerance (see 4.2.104) that is the amount of deviation a surface or line shall have from being straight. No element of the surface or line deviate more than a specified tolerance amount from a straight line.

NOTE 1 Figure 115 illustrates the Straightness_tolerance.

NOTE 2 The Straightness_tolerance definition is derived fromISO 1101:2004, 18.1.

The data associated with a Straightness_tolerance are the following:

— affected_plane;

— segment_size.

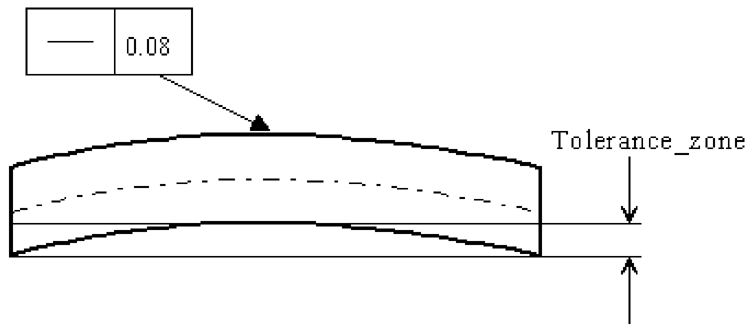


Figure 115 — Straightness_tolerance

4.2.235.1 affected_plane

The affected_plane specifies the plane to apply the tolerance value. The affected_plane is equivalent to a 2D drawing view. The affected_plane need not be specified for a particular Straightness_tolerance. See 4.3.307 for the application assertion.

4.2.235.2 segment_size

The segment_size specifies the length of a surface to apply a tolerance if the Straightness_tolerance is not applied to the total length. The segment_size need not be specified for a particular Straightness_-tolerance.

4.2.236 Surface_profile_tolerance

A Surface_profile_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a uniform boundary or zone along the true profile within which all elements of the surface shall lie.

NOTE 1 Figure 116 illustrates Surface_profile_tolerance.

NOTE 2 The Surface_profile_tolerance definition is derived from ISO 1101:2004, 18.7.

The data associated with a Surface_profile_tolerance are the following:

— geometric_reference.

4.2.236.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.308 for the application assertion.

4.2.237 Supplier_BSU

A Supplier_BSU (supplier basic semantical unit) is a type of BSU (see 4.2.17) that identifies the supplier of a parts library.

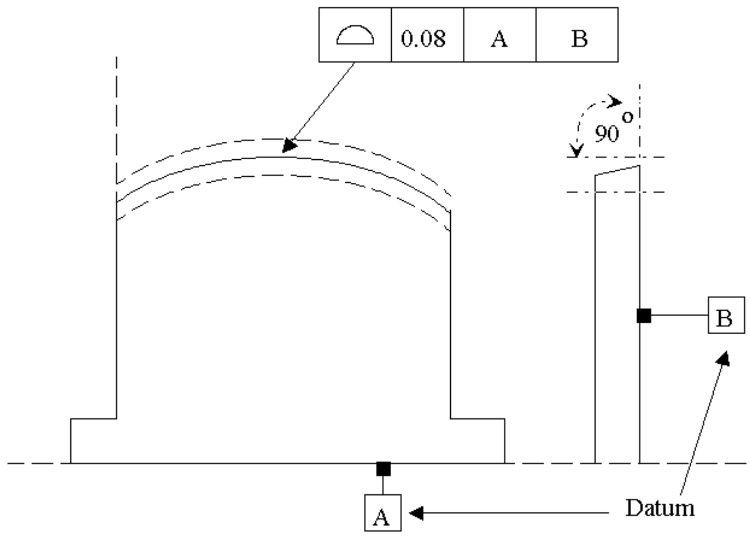


Figure 116 — Surface_profile_tolerance

4.2.238 Surface_property

A Surface_property specifies characteristics of a surface that are elements of the shape of a part. The data associated with a Surface_property are the following:

— property_characteristics;

— surface_finish.

4.2.238.1 property_characteristics

The property_characteristics specifies the parameter to describe the Surface_property. See 4.3.309 for the application assertion.

4.2.238.2 surface_finish

The surface_finish specifies a boolean value that indicates a type of Surface_property is a surface finish. A value of true specifies the Surface_property is a surface finish property, a value of false specifies the Surface_property is for other surface properties.

4.2.239 Symmetry_tolerance

A Symmetry_tolerance is a type of Geometric_tolerance (see 4.2.104) that defines a tolerance zone limited by two parallel planes a distance apart and disposed symmetrically to the median plane with respect to a datum axis or datum plane; or when projected in a plane, limited by two parallel straight lines a distance apart and disposed symmetrically with respect to a datum axis or datum plane.

NOTE 1 Figure 117 illustrates the Symmetry_tolerance.

NOTE 2 The Symmetry_tolerance definition is derived fromISO 1101:2004, 18.14.

The data associated with a Symmetry_tolerance are the following:

— affected_plane;

— geometric_reference.

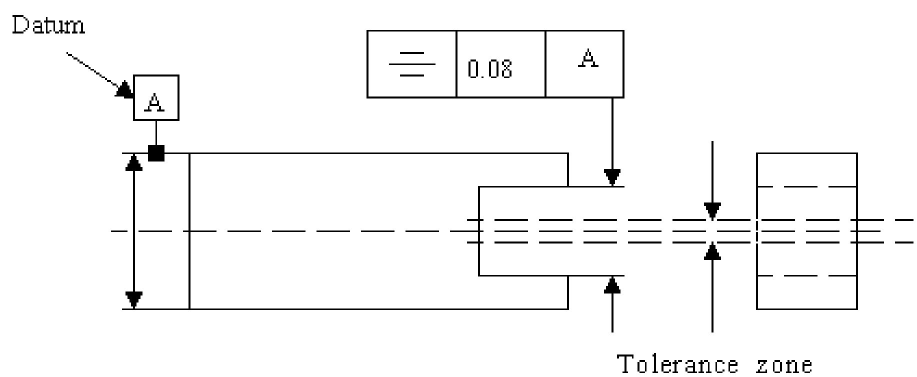


Figure 117 — Symmetry_tolerance

4.2.239.1 affected_plane

The affected_plane specifies the plane to apply the tolerance value. The affected_plane is equivalent to a 2D drawing view. The affected_plane need not be specified for a particular Symmetry_tolerance. See 4.3.311 for the application assertion.

4.2.239.2 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.310 for the application assertion.

4.2.240 Target_area

A Target_area is a type of Datum_target (see 4.2.56) that is an enclosed area bounded by an arbitrary shape required to define a Datum_target. The shape of the Target_area is described explicitly by a set of curves. The data associated with Target_area are the following:

— area_shape.

4.2.240.1 area_shape

The area_shape specifies the physical form of the Target_area shape. See 4.3.312 for the application assertion.

4.2.241 Target_circle

A Target_circle is a type of Placed_target (see 4.2.169) that is an enclosed area bounded by a circle required to define a Datum_target. The origin of the Datum_target is the center of the circle, and the orientation is the x-y plane. The data associated with a Target_circle are the following:

— target_diameter.

4.2.241.1 target_diameter

The target_diameter specifies the diameter value of the Target_circle.

4.2.242 Target_line

A Target_line is a type of Placed_target (see 4.2.169) that is a straight curve. The origin shall be the first end point of the Target_line, the second end point shall be located on the Z-axis at a specified length. The data associated with a Target_line are the following:

— target_length.

4.2.242.1 target_length

The target_length specifies the length value of the Target_line.

4.2.243 Target_point

A Target_point is a type of Placed_target (see 4.2.169) that is a single point. The origin shall be at the Target_point.

4.2.244 Target_rectangle

A Target_rectangle is a type of Placed_target (see 4.2.169) that is an area bounded by four sides with opposite sides equal in length. The center of the rectangle is at the origin. The orientation of the rectangle is with the length along the X-axis and the width along the Y-axis. The data associated a with Target_rectangle are the following:

— target_length;

— target_width.

4.2.244.1 target_length

The target_length specifies the length value of the Target_rectangle.

4.2.244.2 target_width

The target_width specifies the width value of the Target_rectangle.

4.2.245 Tee_profile

A Tee_profile is a type of Open_profile (see 4.2.145) the cross-section of which has the shape of the twentieth letter of the English alphabet in capital form. The first line has a defined length. The second line begins at the midpoint of the first line and is perpendicular to it. The second line extends infinitely. The corners of the Tee_profile need not be blended by a radius. The profile is positioned with the opening in the direction of the Y-axis. The orientation is at a point on the profile the farthest distance from the opening measured along the Y-axis. The X-axis is tangent to the bottom of the profile.

NOTE Figure 118 illustrates the Tee_profile.

The data associated with a Tee_profile are the following:

— cross_bar_depth;

— cross_bar_width;

— depth;

— first_angle;

— first_offset;

— radius;

— second_angle;

— second_offset;

— width.

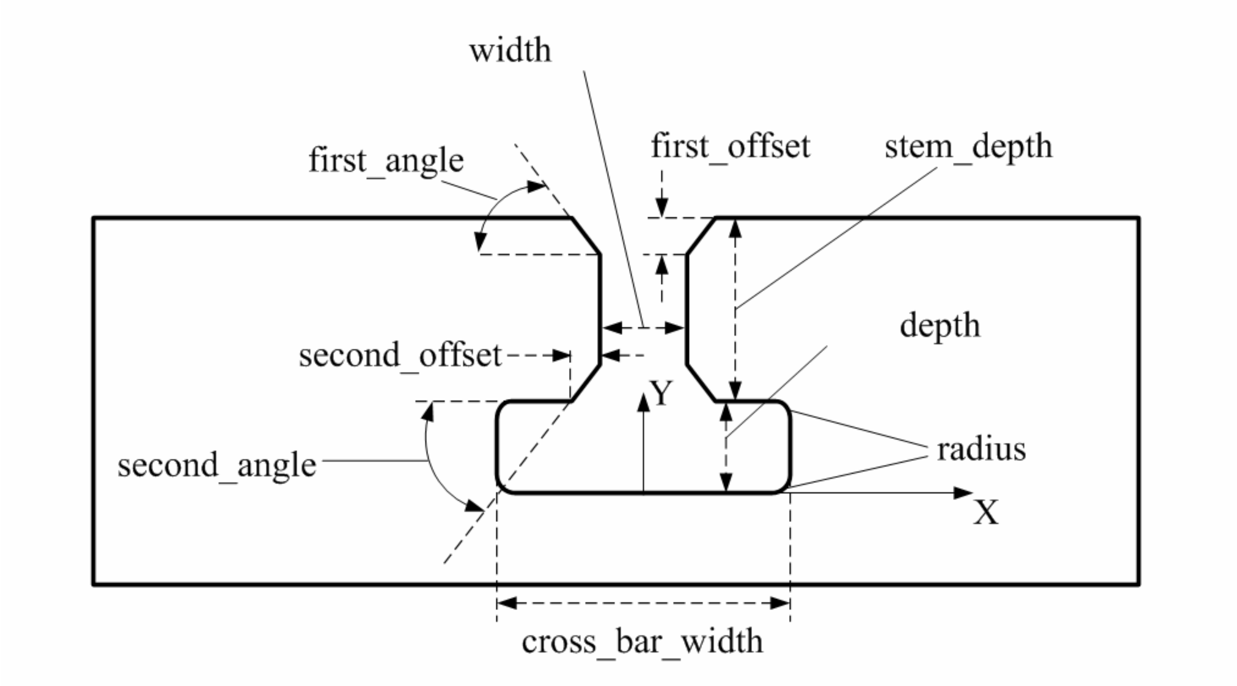


Figure 118 — Tee_profile

4.2.245.1 cross_bar_depth

The cross_bar_depth specifies the depth dimension of the Tee cross bar size. See 4.3.313 for the application assertion.

4.2.245.2 cross_bar_width

The cross_bar_width specifies the width dimension of the Tee cross bar size. See 4.3.313 for the application assertion.

4.2.245.3 depth

The depth specifies the depth dimension of the Tee stem. See 4.3.313 for the application assertion.

4.2.245.4 first_angle

The first_angle specifies the angular measurement for creating a chamfer on the open end of a Tee_-profile. See 4.3.313 for the application assertion.

4.2.245.5 first_offset

The first_offset specifies the distance from the edge of the Tee stem to create a chamfer on the open end of a Tee_profile. See 4.3.313 for the application assertion.

4.2.245.6 radius

The radius specifies the arc size for blending the sides of a Tee_profile cross bar. See 4.3.313 for the application assertion.

4.2.245.7 second_angle

The second_angle specifies the angular measurement for creating a chamfer between the stem and the cross bar parts of a Tee_profile. See 4.3.313 for the application assertion.

4.2.245.8 second_offset

The second_offset specifies a distance from the edge of the Tee stem to create a chamfer a distance from the edge of a surface to the finish of a chamfer. See 4.3.313 for the application assertion.

4.2.245.9 width

The width specifies the width dimension of the Tee stem. See 4.3.313 for the application assertion.

4.2.246 Thread

A Thread is a type of Machining_feature (see 4.2.126) that is a ridge of uniform section on the form of a helix on the external or internal surface of a cylinder. Each Thread is either a Catalogue_thread (see 4.2.21) or a Defined_thread (see 4.2.61).

NOTE 1 Figure 119 illustrates the Thread and Figure 12, Figure 38 illustrates Thread attributes.

NOTE 2 Threads can be used to screw parts together.

NOTE 3 An outside thread might be on a Outer_round, an inside thread might be in a Round_hole.

The data associated with a Thread are the following:

— applied_shape;

— fit_class;

— fit_class_2;

— form;

— inner_or_outer_thread;

— nominal_size;

— number_of_threads;

— partial_profile;

— qualifier;

— runout;

— thread_hand.

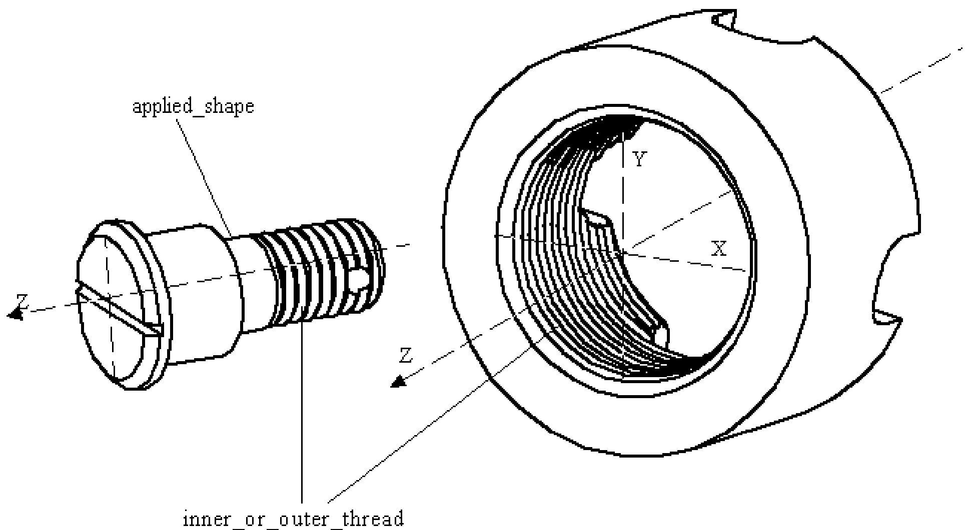


Figure 119 — Thread

4.2.246.1 applied_shape

The applied_shape specifies the physical shape of the Part that will define where the Thread feature will be applied. See 4.3.317 for the application assertion.

4.2.246.2 fit_class

The fit_class specifies the value for the type of fit specification for the thread. These types are distinguished from each other by the amount of tolerance and allowance. The fit_class need not be specified for a particular Thread. See 4.3.314 for the application assertion.

EXAMPLE Examples of ANSI fit class are: 1A, 2A, and 3A which apply to external threads only, and 1B, 2B, and 3B which apply to internal threads only.

4.2.246.3 fit_class_2

The fit_class_2 specifies the value for the type of fit specification for the thread. These types are distinguished from each other by the amount of tolerance and allowance. The fit_class_2 need not be specified for a particular Thread. See 4.3.314 for the application assertion.

EXAMPLE In the case of ISO metric screw threads, where the tolerance class for the major diameter differs to that for the pitch diameter, it is necessary to have two fit classes in order to specify the thread.

4.2.246.4 form

The form specifies the definition of the shape of the thread. Various forms of threads are used to hold parts together, to adjust parts with reference to each other, or to transmit power. See 4.3.314 for the application assertion.

EXAMPLE Examples of form are: metric, square, unified, sharp V, buttress, standard worm, and knuckle.

4.2.246.5 inner_or_outer_thread

An inner_or_outer_thread specifies whether or not the thread is applied as an internal thread or an external thread.

4.2.246.6 nominal_size

The nominal_size specifies the size designation, typically a nominal major diameter, that identifies the thread within a standard thread series. The nominal_size need not be specified for a particular Thread. See 4.3.315 for the application assertion.

4.2.246.7 number_of_threads

The number_of_threads specifies the density of threads per inch when used with English unit of measure and is the thread pitch when used with metric unit of measure. See 4.3.315 for the application assertion.

4.2.246.8 partial_profile

The partial_profile specifies the limitations to be applied on the Thread feature. See 4.3.316 for the application assertion.

4.2.246.9 qualifier

The qualifier specifies additional text information that describes a Thread. The qualifier need not be specified for a particular Thread. See 4.3.314 for the application assertion.

4.2.246.10 runout

The runout specifies part of a Thread with an incomplete thread form. The runout need not be specified for a particular Thread. The runout need not be specified for a particular Thread. See 4.3.318 for the application assertion.

4.2.246.11 thread_hand

The thread_hand specifies a description of whether the thread is right or left handed. When viewed toward an end, a right hand winds in a clockwise direction and a left hand winds in a counterclockwise direction. See 4.3.314 for the application assertion.

4.2.247 Thread_runout

A Thread_runout is a portion of a thread where the thread form is incomplete. It is usually characterised by an approximately linear variation in thread depth from full to nothing, over a given length. A Thread_runout is typically a by-product of the thread cutting process, but may be a design feature of functional importance.

EXAMPLE An example of a thread runout is when used as a locking mechanism for a plain stud.

The data associated with a Thread_runout are the following:

— included_or_extra;

— length_of_runout;

— pitch_or_dimension.

4.2.247.1 included_or_extra

The value of included_or_extra specifies whether or not the thread effective_length is inclusive or exclusive of the length of the runout (length_of_runout). A value of true specifies the runout length as included within the thread effective_length. A value of false specifies the runout length as additional to the thread effective_length.

4.2.247.2 length_of_runout

The length_of_runout specifes the length of the runout. See 4.3.319 for the application assertion.

4.2.247.3 pitch_or_dimension

The pitch_or_dimension specifies whether the length_of_runout is quantified by a dimension or a number of thread pitches. A value of true specifies the runout length is specified as a number of pitches. A value of false specifies the runout length is specified by a dimension.

4.2.248 Thickness_tolerance

A Thickness_tolerance is a type of Size_tolerance (see 4.2.223) 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.

4.2.248.1 path

The path specifies the path along which the Thickness_dimension is applied or measured. See 4.3.320 for the application assertion.

4.2.249 Through_bottom_condition

A Through_bottom_condition is a selection type of Hole_bottom_condition_select that shall pass through two faces of a part; the depth is specified by the feature. The Through_bottom_condition length is specified by the size of the Hole feature.

NOTE Figure 120 illustrates the Through_bottom_condition.

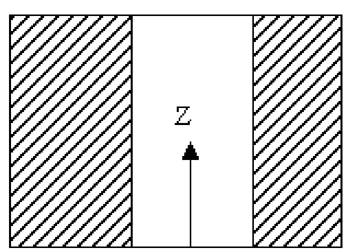


Figure 120 — Through_bottom_condition

4.2.250 Through_pocket_bottom_condition

A Through_pocket_bottom_condition is a pocket that passes through two faces of a part; the depth is defined by the pocket_depth (see 4.2.177.4) of the Pocket feature.

NOTE Figure 61 illustrates Through_pocket_bottom_condition.

4.2.251 Through_profile_floor

A Through_profile_floor is a Shape_profile which passes through two faces of a part; the depth is specified by the size of the feature.

4.2.252 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 4.2.69.3) shall be a tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.143.2) shall be a tolerance value. There shall be a qualifier that describes the tolerance context. The data associated with a Tolerance_limit are the following:

— limit_qualifier.

4.2.252.1 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 4.2.251.1.1 to 4.2.251.1.3 for the definition of each allowable value for limit_qualifier.

4.2.251.1.1 maximum:
the upper limit on a dimension.

4.2.251.1.2 minimum:
the lower limit on a dimension.

4.2.251.1.3 User_definied:
a limit type specified by the user.

4.2.253 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 4.2.69.3) may be a nominal tolerance value. When applied to a Numeric_parameter_with_tolerance, the parameter_value (see 4.2.143.2) may be a nominal tolerance value. The data associated with a Tolerance_range are the following:

— lower_range;

— significant_digits;

— upper_range.

4.2.253.1 lower_range

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

4.2.253.2 significant_digits

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

4.2.253.3 upper_range

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

4.2.254 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. The data associated with a Tolerance_value are the following:

— defined_value.

4.2.254.1 defined_value

The defined_value specifies the tolerance deviation value. See 4.3.321, 4.3.322, 4.3.323 and 4.3.324 for the application assertions.

4.2.255 Tolerance_zone

A Tolerance_zone is an area where all points of the geometric element that have tolerances shall be contained.

EXAMPLE A point, line, surface, or plane are examples for geometric elements that have tolerances zones.

The data associated with a Tolerance_zone are the following:

— common_zone;

— extended_shape;

— form_type;

— zone_definition.

4.2.255.1 common_zone

The common_zone specifies a boolean value that indicates if a Tolerance_zone is applied to more than one geometric tolerance. A TRUE value would indicate a common zone.

4.2.255.2 extended_shape

The extended_shape specifies the extension of a feature for the purpose of creating the Tolerance_zone. The extended_shape need not be specified for a particular Tolerance_zone. See 4.3.326 for the application assertion.

4.2.255.3 form_type

The form_type specifies the shape of the Tolerance_zone.

EXAMPLE 'Cylindrical', 'parallelepiped', 'spherical' are examples for form_type.

4.2.255.4 zone_definition

The zone_definition specifies the defining boundaries for a Tolerance_zone. See 4.3.325 for the application assertion.

4.2.256 Tolerance_zone_definition

A Tolerance_zone_definition is the boundaries of a Tolerance_zone. Each Tolerance_zone shall be defined by at least one shape and may be defined with two shapes.

The data associated with a Tolerance_zone_definition are the following:

— first_element;

— second_element.

4.2.256.1 first_element

The first_element specifies a geometric shape for defining the boundary for the Tolerance_zone_-definition. A second element specifies a second geometric shape. See 4.3.327 for the application assertion.

4.2.256.2 second_element

The second_element specifies the second of two shapes for defining the boundary for the Tolerance_-zone_definition. The second_element need not be specified for a particular Tolerance_zone_definition. See 4.3.327 for the application assertion.

4.2.257 Total_runout_tolerance

A Total_runout_tolerance is a type of Geometric_tolerance (see 4.2.104) that is a compound tolerance used to control the functional relationship of one or more surfaces of a part to a datum axis. The types of surfaces controlled by Total_runout_tolerance tolerances include those surfaces constructed around a datum axis and those constructed at right angles to a datum axis. Surfaces shall be within the tolerance when the part is rotated about the datum axis.

NOTE 1 Figure 121illustrates the Total_runout_tolerance.

NOTE 2 The Total_runout_tolerance definition is derived from ISO 1101:2004, 18.15.

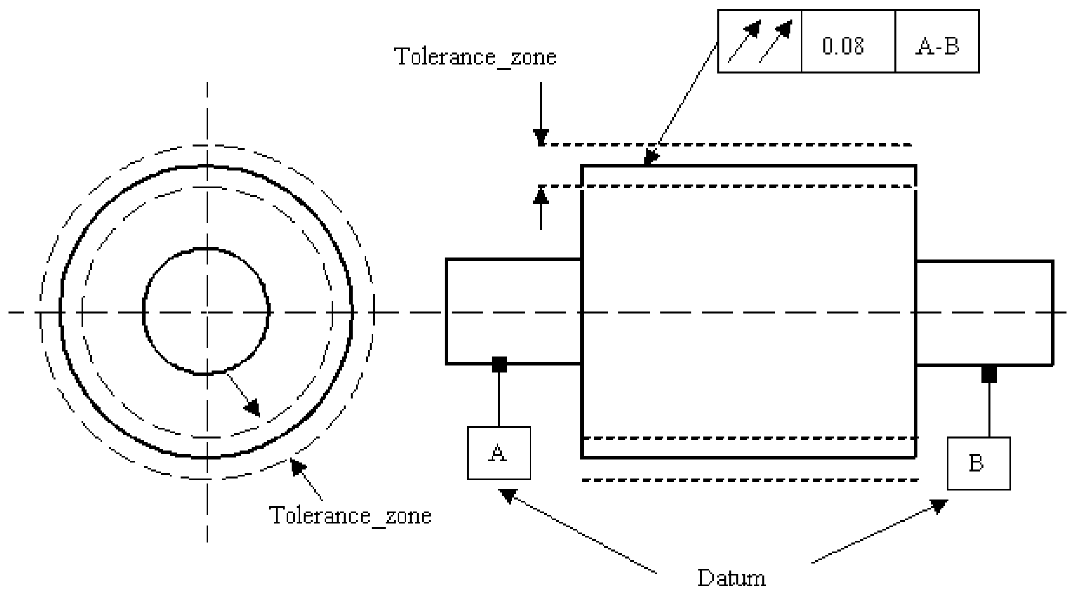


Figure 121 — Total_runout_tolerance

The data associated with a Total_runout_tolerance are the following:

— geometric_reference;

— runout.

4.2.257.1 geometric_reference

The geometric_reference specifies the datum to which the tolerance is related. See 4.3.328 for the application assertion.

4.2.257.2 runout

The runout specifies the direction to control a runout tolerance. If the angle is specified the runout tolerance applies in this angle which is fixed with respect to the datum axis. The runout need not be specified for a particular Total_runout_tolerance.

4.2.258 Transition_feature

A Transition_feature is a type of Manufacturing_feature (see 4.2.129) that is a transition area between two surfaces. This feature differs from Machining_feature objects in that it requires no orientation for placement. Each Transition_feature is either a Chamfer (see 4.2.22), an Edge_round (see 4.2.73), or a Fillet (see 4.2.81).

4.2.259 Turned_knurl

A Turned_knurl is a type of Knurl (see 4.2.114) that is a scoring pattern consisting of a series of shallow cuts on a cylindrical surface. Each Turned_knurl is either a Diagonal_knurl (see 4.2.64), a Diamond_knurl (see 4.2.67), or a Straight_knurl (see 4.2.234).

NOTE 1 Figure 121 illustrates the Turned_knurl.

NOTE 2 A knurl may be used to aid in gripping a part.

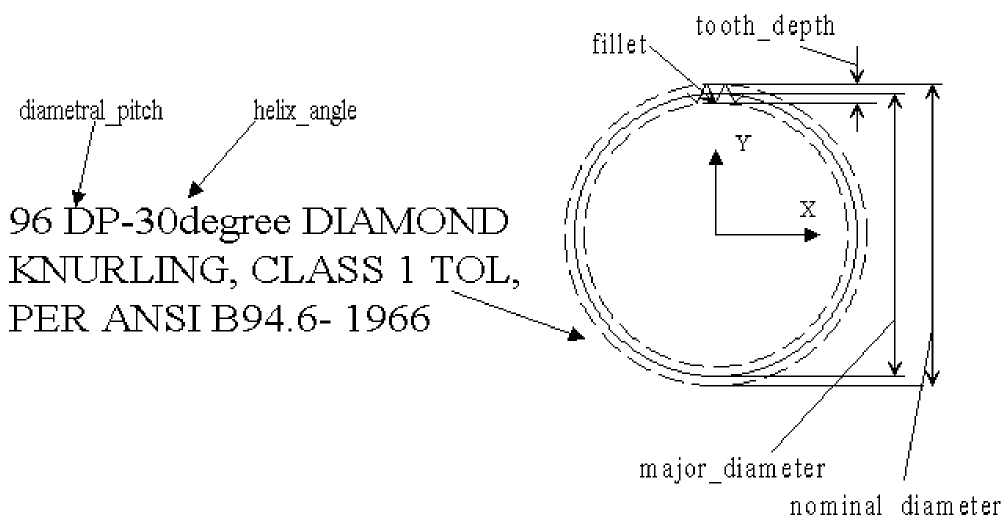


Figure 122 — Turned_knurl

The data associated with a Turned_knurl are the following:

— diametral_pitch;

— major_diameter;

— nominal_diameter;

— number_of_teeth;

— root_fillet;

— tooth_depth.

4.2.259.1 diametral_pitch

The diametral_pitch specifies the ratio of the number of teeth in the circumference to the nominal diameter. See 4.3.329 for the application assertion.

4.2.259.2 major_diameter

The major_diameter specifies size of the part before a knurl is applied to it. See 4.3.329 for the application assertion.

4.2.259.3 nominal_diameter

The nominal_diameter specifies the size of the part after a knurl has been applied. See 4.3.329 for the application assertion.

4.2.259.4 number_of_teeth

The number_of_teeth specifies the number of teeth in the circumference produced on the part surface. The number_of_teeth need not be specified for a particular Turned_knurl. See 4.3.329 for the application assertion.

4.2.259.5 root_fillet

The root _fillet specifies the dimension of a radius between teeth on a knurling tool. See 4.3.329 for the application assertion.

4.2.259.6 tooth_depth

The tooth_depth specifies the depth from the crest of a tooth to the point where two teeth intersect. See 4.3.329 for the application assertion.

4.2.260 Vee_profile

A Vee_profile is a type of Open_profile (see 4.2.145) that is a shape bounded by two lines that connect at a point and extends infinitely. The enclosed angle is less than 180 degrees. The intersection of the two lines need not be blended with a radius. The profile is positioned with the opening in the direction of the Y axis. The Y-axis intersects the angle between the two sides.

NOTE Figure 123 illustrates the Vee_profile.

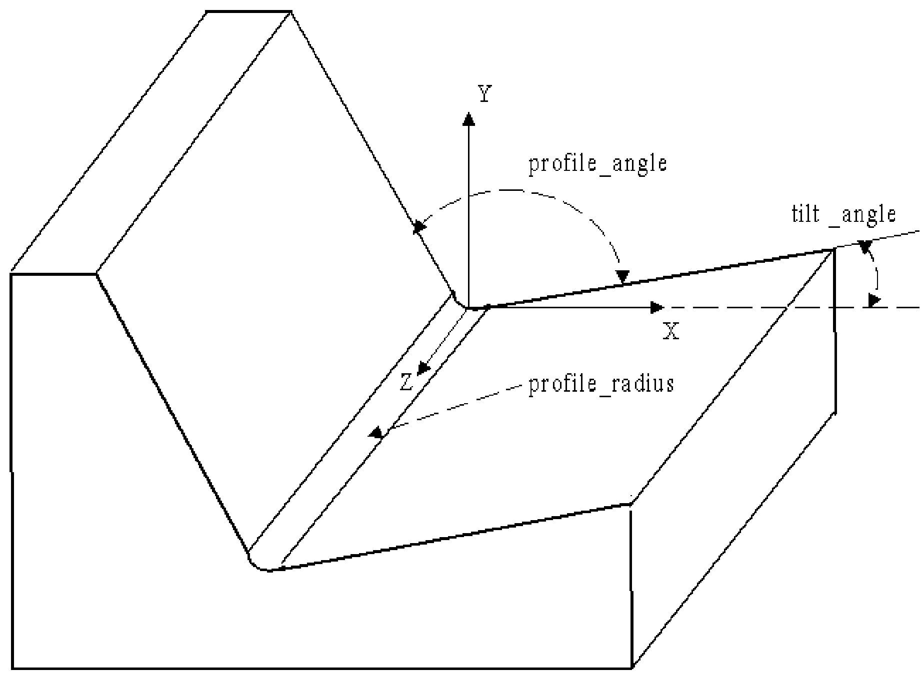


Figure 123 — Vee_profile

The data associated with a Vee_profile are the following:

— first_length;

— profile_angle;

— profile_radius;

— second_length;

— tilt_angle.

4.2.260.1 first_length

The first_length indicates the distance, as measured from the profile origin, along the side of the vee located by the tilt_angle parameter. See 4.3.330 for the application assertion.

4.2.260.2 profile_angle

The profile_angle specifies the size of the angle between the two sides of the Vee_profile. The angle shall be greater than 0 and not more than 180 degrees. See 4.3.330 for the application assertion.

4.2.260.3 profile_radius

The profile_radius specifies the size of the blend radius at the point of the V, or where the two sides come together. See 4.3.330 for the application assertion.

4.2.260.4 second_length

The second_side_length indicates the distance, as measured from the profile origin, along the side of the vee located by the sum of the tilt_angle and profile_angle. See 4.3.330 for the application assertion.

4.2.260.5 tilt_angle

The tilt_angle specifies the size of the angle between one side of the Vee_profile and the x-axis of the local coordinate system that defines the Vee_profile orientation on the part. See 4.3.330 for the application assertion.

4.2.261 Width_dimension

A Width_dimension is a type of Size_tolerance (see 4.2.223) 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.

4.2.261.1 path

The path specifies the path along which the Width_dimension is applied or measured. See 4.3.331 for the application assertion.

4.2.262 Woodruff_slot_end_type

A Woodruff_slot_end_type is a type of Slot_end_type (see 4.2.225) that is an end condition of a slot that shall be a radius tangent to the Slot bottom, and curved upward about an axis.

NOTE Figure 124 illustrates the Woodruff_slot_end_type.

The data associated with a Woodruff_slot_end_type are the following:

— radius.

4.2.262.1 radius

The radius specifies the size of the radius swept about an axis, tangent to the Slot bottom and the end of a Slot. See 4.3.332 for the application assertion.

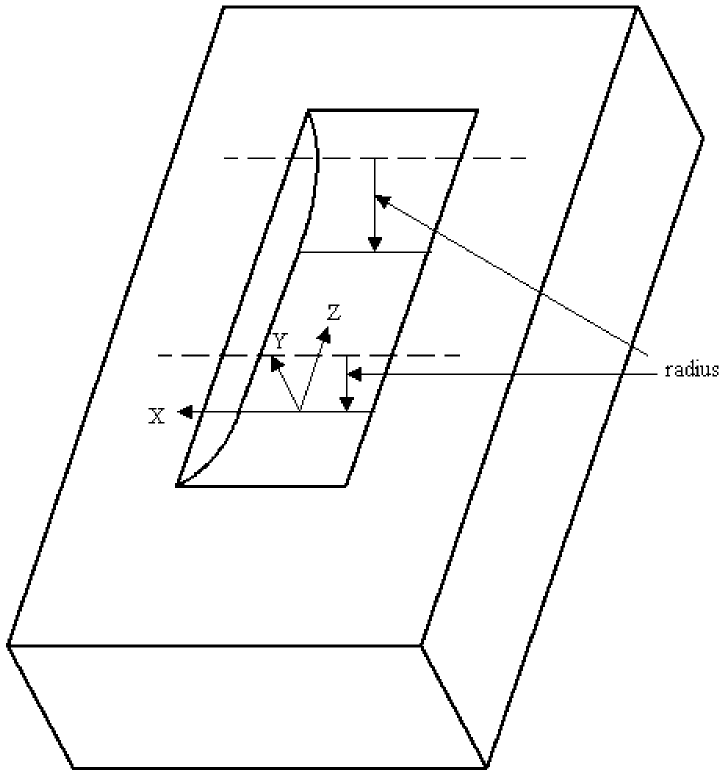


Figure 124 — Woodruff_slot_end_type

4.3 Application assertions

4.3.1 General

This subclause specifies the application assertions for the Mechanical product definition for process planning using machining features application protocol. Application assertions specify the relationships between application objects, the cardinality of the relationships, and the rules required for the integrity and validity of the application objects and UoFs. The application assertions and their definitions are given below.

4.3.2 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.

4.3.3 Angle_taper to Numeric_parameter

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

4.3.4 Angular_dimension_tolerance to Orientation

Each Angular_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 Angular_dimension_tolerance objects.

4.3.5 Angularity_tolerance to Datum

Each Angularity_tolerance has the geometric_reference defined by exactly one Datum. Each Datum is the geometric_reference for zero, one, or many Angularity_tolerance objects.

4.3.6 Approval to Person_in_organization

Each Approval has the approval_authority defined by one or many Person_in_organization objects. Each Person_in_organization is the approval_authority for zero, one, or many Approval objects.

4.3.7 Bevel_gear to Numeric_parameter

Each Bevel_gear has the root_angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the root_angle for zero, one, or many Bevel_gear objects.

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

4.3.8 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.

4.3.9 Boss to Boss_top_condition

Each Boss has the top_condition defined by exactly one Boss_top_condition. Each Boss_top_condition defines the top_condition for zero, one or many Boss objects.

4.3.10 Boss to Linear_path

Each Boss has the boss_height defined by exactly one Linear_path. Each Linear_path defines the boss_height for zero, one, or many Boss objects.

4.3.11 Boss to Numeric_parameter

Each Boss has the fillet_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the fillet_radius for zero, one, or many Boss objects.

4.3.12 Boss_top_condition to Numeric_parameter

Each Boss has the top_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the top_radius for zero, one, or many Boss objects.

4.3.13 B-rep_model_element to B-rep_model

Each B-rep_model_element defines an element of exactly one B-rep_model. Each B-rep_model defines the element for zero, one, or many B-rep_model_element objects.

4.3.14 B-rep_shape_aspect_representation to B-rep_model

Each B-rep_shape_aspect_representation has the shape_definition defined by exactly one B-rep_model. Each B-rep_model defines the shape_definition for zero, one, or many B-rep_shape_aspect_-representation objects.

4.3.15 B-rep_shape_representation to B-rep_model

Each B-rep_shape_representation has the shape_definition defined by exactly one B-rep_model. Each B-rep_model defines the shape_definition for zero, one, or many B-rep_shape_representation objects.

4.3.16 Catalogue_gear to Specification

Each Catalogue_gear has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_gear objects.

4.3.17 Catalogue_knurl to Specification

Each Catalogue_knurl has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_knurl objects.

4.3.18 Catalogue_marking to Specification

Each Catalogue_marking has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_marking objects.

4.3.19 Catalogue_thread to Specification

Each Catalogue_thread has documentation defined by exactly one Specification. Each Specification defines the documentation for zero, one, or many Catalogue_thread objects.

4.3.20 Catalogue_thread to Numeric_parameter

Each Catalogue_thread has the major_diameter defined by zero or one Numeric_parameter. Each Numeric_parameter defines the major_diameter for zero, one, or many Catalogue_thread objects.

4.3.21 Chamfer to Face_shape_element

Each Chamfer has the chamfer_face identified by exactly one Face_shape_element. Each Face_shape_-element is the chamfer_face shape for zero, one, or many Chamfer objects.

4.3.22 Chamfer to First_offset

Each Chamfer has the first_face_offset defined by exactly one First_offset. Each First_offset defines the first_face_offset for zero, one, or many Chamfer objects.

4.3.23 Chamfer to Second_chamfer_offset

Each Chamfer has the second_face_offset defined by exactly one Second_chamfer_offset. Each Second_chamfer_offset defines the second_face_offset for zero, one, or many Chamfer objects.

4.3.24 Chamfer_angle to Numeric_parameter

Each Chamfer_angle has the angle_amount defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the angle_amount for zero, one, or many Chamfer_angle objects.

4.3.25 Circular_boss to Circular_closed_profile

Each Circular_boss has the circular_profile defined by exactly one Circular_closed_profile. Each Circular_closed_profile defines the circular_profile for zero, one, or many Circular_boss objects.

4.3.26 Circular_boss to Angle_taper

Each Circular_boss has the change_in_diameter defined by zero or one Angle_taper. Each Angle_taper defines the change_in_diameter for zero, one, or many Circular_boss objects.

NOTE This assertion is established through Taper_select.

4.3.27 Circular_boss to Diameter_taper

Each Circular_boss has the change_in_diameter defined by zero or one Diameter_taper. Each Diameter_-taper defines the change_in_diameter for zero, one, or many Circular_boss objects.

NOTE This assertion is established through Taper_select.

4.3.28 Circular_boss to Directed_taper

Each Circular_boss has the change_in_diameter defined by zero or one Directed_taper. Each Directed_-taper defines the change_in_diameter for zero, one, or many Circular_boss objects.

NOTE This assertion is established through Taper_select.

4.3.29 Circular_closed_profile to Numeric_parameter

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

4.3.30 Circular_closed_shape_profile to Circular_closed_profile

Each Circular_closed_shape_profile has the closed_boundary defined by exactly one Circular_closed_-profile. Each Circular_closed_profile defines the closed_boundary for zero, one, or many Circular_-closed_shape_profile objects.

4.3.31 Circular_cutout to Circular_closed_profile

Each Circular_cutout has the closed_boundary defined by exactly one Circular_closed_profile. Each Circular_closed_profile defines the closed_boundary for zero, one, or many Circular_cutout objects.

4.3.32 Circular_offset_pattern to Numeric_parameter

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

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

4.3.33 Circular_omit_pattern to Numeric_parameter

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

4.3.34 Circular_path to Numeric_parameter

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

4.3.35 Circular_pattern to Circular_offset_pattern

Each Circular_pattern has the relocated_base_feature defined by zero, one, or many Circular_offset_-pattern objects. Each Circular_offset_pattern defines the relocated_base_feature for one or more Circular_pattern objects.

4.3.36 Circular_pattern to Circular_omit_pattern

Each Circular_pattern has the missing_base_feature defined by zero, one, or many Circular_omit_pattern objects. Each Circular_omit_pattern defines the missing_base_feature for one or more Circular_pattern objects.

4.3.37 Circular_pattern to Numeric_parameter

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

Each Circular_pattern has the base_feature_diameter defined by zero or. one Numeric_parameter. Each Numeric_parameter defines the base_feature_diameter for zero, one, or many Circular_pattern objects.

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

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

4.3.38 Circular_runout_tolerance to Datum

Each Circular_runout_tolerance has the geometric_reference defined by one or two Datum objects. Each Datum defines the geometric_reference for zero, one, or many Circular_runout_tolerance objects.

4.3.39 Class_BSU to Supplier_BSU

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

4.3.40 Closed_slot to Complete_circular_path

Each Closed_slot has the course_of_travel defined by exactly one Complete_circular_path. Each Complete_circular_path defines the course_of_travel for zero, one, or many Closed_slot objects.

NOTE This assertion is established through end_type_select.

4.3.41 Closed_slot to General_path

Each Closed_slot has the course_of_travel defined by exactly one General_path. Each General_path defines the course_of_travel for zero, one, or many Closed_slot objects.

NOTE This assertion is established through end_type_select.

4.3.42 Closed_slot to Loop_slot_end_type

Each Closed_slot has the end_conditions defined by exactly two Loop_slot_end_type objects. Each Loop_slot_end_type defines the end_condition for zero, one, or many Closed_slot objects.

4.3.43 Compound_datum to Datum_feature

Each Compound_datum has the element defined by two or more Datum_feature objects. Each Datum_feature defines the element for zero, one, or many Compound_datum objects.

4.3.44 Compound_feature to Compound_feature_element

Each Compound_feature has the element defined by two or more Compound_feature_element objects. Each Compound_feature_element defines the element for zero, one, or many Compound_feature objects.

4.3.45 Compound_feature_element to Machining_feature

Each Compound_feature_element has the element defined by exactly one Machining_feature. Each Machining_feature defines the element for zero, one, or many Compound_feature_element objects.

NOTE This assertion is established through Compound_feature_select.

4.3.46 Compound_feature_element to Transition_feature

Each Compound_feature_element has the element defined by exactly one Transition_feature. Each Transition_feature defines the element for zero, one, or many Compound_feature_element objects.

NOTE This assertion is established through Compound_feature_select.

4.3.47 Compound_feature_relationship to Compound_feature_element

Each Compound_feature_relationship has the successor defined by exactly one Compound_feature_-element. Each Compound_feature_element defines the successor for zero, one, or many Compound_-feature_relationship objects.

Each Compound_feature_relationship has the predecessor defined by exactly one Compound_feature_-element. Each Compound_feature_element defines the predecessor for zero, one, or many Compound_-feature_relationship objects.

4.3.48 Concentricity_tolerance to Datum

Each Concentricity_tolerance has the geometric_reference defined by exactly one Datum. Each Datum is the geometric_reference for zero, one, or many Concentricity_tolerance objects.

4.3.49 Conical_hole_bottom to Numeric_parameter

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

Each Conical_hole_bottom has the tip_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the tip_radius for zero, one, or many Conical_hole_bottom objects.

4.3.50 Constant_radius_edge_round to Numeric_parameter

Each Constant_radius_edge_round has the first_face_offset defined by zero or one Numeric_parameter. Each Numeric_parameter defines the first_face_offset for zero, one, or many Constant_radius_-edge_round objects.

Each Constant_radius_edge_round has the second_face_offset defined by zero or one Numeric_-parameter. Each Numeric_parameter defines the second_face_offset for zero, one, or many Constant_-radius_edge_round objects.

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

4.3.51 Constant_radius_fillet to Numeric_parameter

Each Constant_radius_fillet has the first_face_offset defined by zero or one Numeric_parameter. Each Numeric_parameter defines the first_face_offset for zero, one, or many Constant_radius_fillet objects.

Each Constant_radius_fillet has the second_face_offset defined by zero or one Numeric_parameter. Each Numeric_parameter defines the second_face_offset for zero, one, or many Constant_radius_fillet objects.

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

4.3.52 Counterbore_hole to Round_hole

Each Counterbore_hole has the smaller_hole defined by exactly one Round_hole. Each Round_hole defines the smaller_hole for zero, one, or many Counterbore_hole objects.

Each Counterbore_hole has the larger_hole defined by exactly one Round_hole. Each Round_hole defines the larger_hole for zero, one, or many Counterbore_hole objects.

4.3.53 Countersunk_hole to Round_hole

Each Countersunk_hole has the constant_diameter_hole defined by exactly one Round_hole. Each Round_hole defines the constant_diameter_hole for zero, one, or many Countersunk_hole objects.

Each Countersunk_hole has the tapered_hole defined by exactly one Round_hole. Each Round_hole defines the taper hole for zero, one, or many Countersunk_hole objects.

4.3.54 Customer_order to Ordered_part

Each Customer_order has the quantity_ordered defined by one or many Ordered_part objects. Each Ordered_part defines the quantity_ordered for zero or one Customer_order.

4.3.55 Customer_order to Person_in_organization

Each Customer_order has the customer defined by exactly one Person_in_organization. Each Person_in_organization defines the customer for zero, one, or many Customer_order objects.

4.3.56 Customer_order to Project_order

Each Customer_order has the initiated_order defined by one or many Project_order objects. Each Project_order defines the initiated_order for zero or one Customer_order.

4.3.57 Cutout to Through_pocket_bottom_condition

Each Cutout has the bottom_condition defined by exactly one Through_pocket_bottom_condition. Each Through_pocket_bottom_condition defines the bottom_condition for zero, one, or many Cutout objects.

4.3.58 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.

4.3.59 Datum_feature to Material_condition_modifier

Each Datum has the modifier defined by zero or one Material_condition_modifier. Each Material_condition_modifier defines the modifier for zero, one, or many Datum objects.

4.3.60 Datum_feature to Datum_target_set

Each Datum_feature has the datum_representation defined by exactly one Datum_target_set. Each Datum_target_set defines the datum_representation for zero, one, or many Datum_feature objects.

NOTE This assertion is established through Datum_representation_select.

4.3.61 Datum_feature to Shape_element

Each Datum_feature has the datum_representation defined by exactly one Shape_element. Each Shape_-element defines the datum_representation for zero, one, or many Datum_feature objects.

NOTE This assertion is established through Datum_representation_select.

4.3.62 Datum_target_set to Datum_target

Each Datum_target_set has the target_shape defined by one or many Datum_target objects. Each Datum_target defines the target_shape for zero, one, or many Datum_target_set objects.

4.3.63 Defined_marking to Descriptive_parameter

Each Defined_marking has the special_instructions defined by zero or one Descriptive_parameter. Each Descriptive_parameter defines special_instructions for zero, one, or many Defined_marking objects.

Each Defined_marking has the font_name defined by zero or one Descriptive_parameter. Each Descriptive_parameter defines font_name for zero, one, or many Defined_marking objects.

4.3.64 Defined_marking to Numeric_parameter

Each Defined_marking has the character_height defined by zero or one Numeric_parameter. Each Numeric_parameter defines the character_height for zero, one, or many Defined_marking objects.

Each Defined_marking has the character_spacing defined by zero or one Numeric_parameter. Each Numeric_parameter defines the character_spacing for zero, one, or many Defined_marking objects.

4.3.65 Defined_thread to Numeric_parameter

Each Defined_thread has the crest defined by zero or one Numeric_parameter. Each Numeric_parameter defines the crest for zero, one, or many Defined_thread objects.

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

Each Defined_thread has the minor_diameter defined by zero or one Numeric_parameter. Each Numeric_parameter defines the minor_diameter for zero, one, or many Defined_thread objects.

Each Defined_thread has the pitch_diameter defined exactly one Numeric_parameter. Each Numeric_parameter defines the pitch_diameter for zero, one, or many Defined_thread objects.

4.3.66 Design_exception_notice to Engineering_change_proposal

Each Design_exception_notice issues zero, one, or many Engineering_change_proposal objects. Each Engineering_change_proposal formalized exactly one Design_exception_notice.

4.3.67 Design_exception_notice to Part

Each Design_exception_notice has the discrepant_part defined by one or many Part objects. Each Part defines the discrepant_part for zero, one, or many Design_exception_notice objects.

4.3.68 Diagonal_knurl to Descriptive_parameter

Each Diagonal_knurl has the helix_hand defined by exactly one Descriptive_parameter. Each Descriptive_parameter defines the helix_hand for zero, one, or many Diagonal_knurl objects.

4.3.69 Diagonal_knurl to Numeric_parameter

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

4.3.70 Diameter_taper to Numeric_parameter

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

4.3.71 Diamond_knurl to Numeric_parameter

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

4.3.72 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.

4.3.73 Directed_taper to Direction_element

Each Directed_taper has the direction defined by exactly one Direction_element. Each Direction_element defines the direction for zero, one, or many Directed_taper objects.

4.3.74 Directed_taper to Numeric_parameter

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

4.3.75 Distance_along_curve_tolerance to Shape_aspect

Each Distance_along_curve_tolerance_tolerance has a path defined by exactly one Shape_aspect. Each Shape_aspect defines the path for zero, one, or many Distance_along_curve_tolerance objects.

4.3.76 Edge_round to Face_shape_element

Each Edge_round has the edge_round_face defined by exactly one Face_shape_element. Each Face_-shape_element defines the edge_round_face for zero, one, or many Edge_round objects.

Each Edge_round has the first_face_shape defined by exactly one Face_shape_element. Each Face_-shape_element defines the first_face_shape for zero, one, or many Edge_round objects.

Each Edge_round has the second_face_shape defined by exactly one Face_shape_element. Each Face_-shape_element defines the second_face_shape for zero, one, or many Edge_round objects.

4.3.77 Engineering_change_order to Part

Each Engineering_change_order has the new_version defined by one or many Part objects. Each Part defines the new_version_changes by zero or one Engineering_change_order.

4.3.78 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.

4.3.79 Explicit_base_shape_representation to B-rep_shape_representation

Each Explicit_base_shape_representation has the B-rep_form defined by exactly one B-rep_shape_-representation. Each B-rep_shape_representation defines the B-rep_form for zero, one, or many Explicit_base_shape_representation objects.

4.3.80 Externally_defined_representation to Orientation

Each Externally_defined_representation has the location_placement defined by exactly one orientation object. Each Orientation defines the location_placement for zero, one, or many Externallt_defined_-representation objects.

4.3.81 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.

4.3.82 Externally_defined_size_dimension to Shape_aspect

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

4.3.83 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.

4.3.84 Face_shape_element_relationship to Face_shape_element

Each Face_shape_element_relationship has the predecessor defined by exactly one Face_shape_element. Each Face_shape_element defines the predecessor for zero, one, or many Face_shape_element_-relationship objects.

Each Face_shape_element_relationship has the successor defined by exactly one Face_shape_element. Each Face_shape_element defines the successor for zero, one, or many Face_shape_element_relationship objects.

4.3.85 Fillet to Face_shape_element

Each Fillet has the fillet_face defined by exactly one Face_shape_element. Each Face_shape_element defines the fillet_face for zero, one, or many Fillet objects.

Each Fillet has the first_face_shape defined by exactly one Face_shape_element. Each Face_shape_element defines the first_face_shape for zero, one, or many Fillet objects.

Each Fillet has the second_face_shape defined by exactly one Face_shape_element. Each Face_shape_element defines the second_face_shape for zero, one, or many Fillet objects.

4.3.86 First_offset to Face_shape_element

Each First_offset has the face_shape defined by exactly one Face_shape_element. Each Face_shape_element defines the face_shape for zero, one, or many First_offset objects.

4.3.87 First_offset to Numeric_parameter

Each First_offset has the offset_amount defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the offset_amount for zero, one, or many First_offset objects.

4.3.88 Flat_slot_end_type to Numeric_parameter

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

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

4.3.89 Flat_with_radius_hole_bottom to Numeric_parameter

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

4.3.90 Flat_with_taper_hole_bottom to Numeric_parameter

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

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

4.3.91 Gear to Shape_element

Each Gear has the applied_shape defined by exactly one Shape_element. Each Shape_element defines the applied_shape for zero, one, or many Gear objects.

4.3.92 Gear to Numeric_parameter

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

Each Gear has the nominal_tooth_depth defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the nominal_tooth_depth for zero, one, or many Gear objects.

Each Gear has the normal_attribute defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the normal_attribute for zero, one, or many Gear objects.

Each Gear has the number_of_teeth defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the number_of_teeth for zero, one, or many Gear objects.

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

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

Each Gear has the root_fillet_radius defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the root_fillet_radius for zero, one, or many Gear objects.

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

4.3.93 General_boss to Closed_profile

Each General_boss has enclosed_boundary defined by exactly one Closed_profile. Each Closed_profile defines the enclosed_boundary for zero, one, or many General_boss objects.

4.3.94 General_boss to Angle_taper

Each General_boss has the change_in_boundary defined by zero or one Angle_taper. Each Angle_taper defines the change_in_boundary for zero, one, or many General_boss objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.95 General_boss to Directed_taper

Each General_boss has the change_in_boundary defined by zero or one Directed_taper. Each Directed_taper defines the change_in_boundary for zero, one, or many General_boss objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.96 General_closed_profile to Path_element

Each General_closed_profile has the closed_profile_shape defined by exactly one Path_element. Each Path_element defines the closed_profile_shape for zero, one, or many General_closed_profile objects.

4.3.97 General_cutout to Profile

Each General_cutout has the boundary defined by exactly one Profile. Each Profile defines the boundary for zero, one, or many General_cutout objects.

4.3.98 General_open_profile to Path_element

Each General_open_profile has enclosed_boundary defined by exactly one Path_element. Each Path_element identifies the enclosed_boundary for zero, one, or many General_open_profile objects.

4.3.99 General_outside_profile to Profile

Each General_outside_profile has boundary defined by exactly one Profile. Each Profile defines outside boundary for zero, one, or many General_outside_profile objects.

4.3.100 General_path to Path_element

Each General_path has the sweep_path defined by exactly one Path_element. Each Path_element defines the sweep_path for zero, one, or many General_path objects.

4.3.101 General_pattern to Orientation

Each General_pattern has the feature_placement defined by one or more Orientation. Each Orientation defines the feature_placement for zero, one, or many General_pattern objects.

4.3.102 General_pocket to Profile

Each General_pocket has the boundary defined by exactly one Profile. Each Profile defines the boundary for zero, one, or many General_pocket objects.

4.3.103 General_pocket to Boss

Each General_pocket has the volume_not_removed defined by zero, one or many Boss objects. Each Protrusion defines the volume_not_removed for zero, one, or many General_pocket objects.

NOTE This assertion is established through Volume_select.

4.3.104 General_pocket to Protrusion

Each General_pocket has the volume_not_removed defined by zero, one or many Protrusion objects. Each Protrusion defines the volume_not_removed for zero, one, or many General_pocket objects.

NOTE This assertion is established through Volume_select.

4.3.105 General_pocket_bottom_condition to Face_shape_element

Each General_pocket_bottom_condition has the floor defined by exactly one Face_shape_element. Each Face_shape_element defines the floor for zero, one, or many General_pocket_bottom_condition objects.

4.3.106 General_pocket_bottom_condition to Numeric_parameter

Each General_pocket_bottom_condition has the floor_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the floor_radius for zero, one, or many General_pocket_bottom_-condition objects.

4.3.107 General_profile_floor to Face_shape_element

Each General_profile_floor has the floor defined by exactly one Face_shape_element. Each Face_-shape_element defines the floor for zero, one, or many General_profile_floor objects.

4.3.108 General_removal_volume to Shape_element

Each General_removal_volume has the removal_volume defined by one or more Shape_element objects. Each Shape_element defines the removal_volume for zero, one, or many General_removal_volume objects.

4.3.109 General_revolution to General_open_profile

Each General_revolution has the outer_edge_shape defined by exactly one General_open_profile. Each General_open_profile defines outer_edge_shape for zero, one, or many General_revolution objects.

4.3.110 General_rib_top_floor to Face_shape_element

Each General_rib_top_floor has the rip_top_face defined by one or more Ordered_face_element objects. Each Ordered_face_element defines the rib_top_floor for zero, one, or many General_rib_top_floor objects.

4.3.111 General_shape_profile to Profile

Each General_shape_profile has the profile_boundary defined by exactly one Profile. Each Profile defines the profile_boundary for zero, one, or many General_shape_profile objects.

4.3.112 General_top_condition to Face_shape_element

Each General_top_condition has the top_face defined by exactly one Face_shape_element. Each Face_shape_element defines the top_face for zero, one, or many General_top_condition objects.

4.3.113 Geometric_tolerance to Material_condition_modifier

Each Geometric_tolerance has the modifier_control defined by zero or one Material_condition_modifier. Each Material_condition_modifier defines the modifier_control for zero, one, or many Geometric_-tolerance objects.

4.3.114 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.

4.3.115 Geometric_tolerance to Tolerance_zone

Each Geometric_tolerance has the zone_definition defined by zero or one Tolerance_zone. Each Tolerance_zone defines the zone_definition for zero, one, or many Geometric_tolerance objects.

4.3.116 Geometric_tolerance_precedence_relationship to Geometric_-tolerance

Each Geometric_tolerance_precedence_relationship has the base_shape_tolerance defined by exactly one Geometric_tolerance. Each Geometric_tolerance defines the base_shape_tolerance for zero, one, or many Geometric_tolerance_precedence_relationship objects.

Each Geometric_tolerance_precedence_relationship has the pattern_shape_tolerance defined by exactly one Geometric_tolerance. Each Geometric_tolerance defines the pattern_shape_tolerance for zero, one, or many Geometric_tolerance_precedence_relationship objects.

4.3.117 Groove to Open_profile

Each Groove has a sweep defined by exactly one Open_profile. Each Open_profile defines the sweep for zero, one, or many Groove objects.

4.3.118 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.

4.3.119 Helical_bevel_gear to Numeric_parameter

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

4.3.120 Helical_gear to Numeric_parameter

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

4.3.121 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.

4.3.122 Implicit_base_shape_representation to Orientation

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

4.3.123 Knurl to Partial_area_definition

Each Knurl has the partial_profile defined by zero or one Partial_area_definition. Each Partial_area_-definition defines the partial_profile for zero, one, or many Knurl objects.

4.3.124 Knurl to Shape_element

Each Knurl has the applied_shape defined by exactly one Shape_element. Each Shape_element defines the applied_shape for zero, one, or many Knurl objects.

4.3.125 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.

4.3.126 Library_part_assignment to Property_value

Each Library_part_assignment has definitional_property_value_pairs defined by zero, one, or many Property_value. Each Property_value defines the definitional_property_value_pairs for zero, one, or many Library_part_assignment objects.

4.3.127 Library_part_assignment to Class_BSU

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

4.3.128 Linear_path to Numeric_parameter

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

4.3.129 Linear_path to Direction_element

Each Linear_path has the direction defined by exactly one Direction_element. Each Direction_element defines the direction for zero, one, or many Linear_path objects.

4.3.130 Linear_profile to Numeric_parameter

Each Linear_profile has the profile_length defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the profile_length for zero, one, or many Linear_profile objects.

4.3.131 Linear_profile_tolerance to Datum

Each Linear_profile_tolerance has the geometric_reference defined by zero, one, two, or three Datum objects. Each Datum defines the geometric_reference for zero, one, or many Linear_profile_tolerance objects.

4.3.132 Linear_profile_tolerance to Orientation

Each Linear_profile_tolerance has an affected_plane defined by zero or one Orientation. Each Orientation defines an affected plane for zero, one, or many Linear_profile_tolerance objects.

4.3.133 Location_dimension_tolerance to Orientation

Each Location_dimension_tolerance has the plane and direction defined by zero or one Shape_aspect. Each Orientation defines the plane and direction for zero, one, or many Location_dimension_tolerance objects.

4.3.134 Location_tolerance to Shape_element

Each Location_tolerance has the termination_shape defined by exactly one Shape_element. Each Shape_element defines the termination_shape for zero, one, or many Location_dimension objects.

Each Location_dimension has the origin_shape defined by exactly one Shape_element. Each Shape_-element defines the origin_shape for zero, one, or many Location_dimension objects.

4.3.135 Machining_feature to Orientation

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

4.3.136 Manufacured_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.

4.3.137 Manufactured_assembly_relationship to Part

Each Manufactured_assembly_relationship has the component defined by exactly one Part. Each Part defines the component for zero, one, or many Manufactured_assembly_relationship objects.

4.3.138 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.

4.3.139 Manufacturing_feature_group to Manufacturing_feature

Each Manufacturing_feature_group has the feature_groups defined by two or more Manufacturing_-feature objects. Each Manufacturing_feature defines the feature_groups for zero, one, or many Manufacturing_feature_group objects.

NOTE This assertion is established through Manufacturing_group_select.

4.3.140 Manufacturing_feature_group to Manufacturing_feature_group

Each Manufacturing_feature_group has the feature_groups defined by two or more Manufacturing_-feature_group objects. Each Manufacturing_feature_group defines the feature_groups for zero, one, or many Manufacturing_feature_group objects.

NOTE This assertion is established through Manufacturing_group_select.

4.3.141 Marking to Descriptive_parameter

Each Marking has text defined by exactly one Descriptive_parameter. Each Descriptive_parameter defines text for zero, one, or many Marking objects.

4.3.142 Marking to Shape_element

Each Marking has an applied_to_shape defined by exactly one Shape_element. Each Shape_element defines the applied_to_shape for zero, one, or many Marking objects.

4.3.143 Material to Material_property

Each Material has the material_characteristics defined by zero, one, or many Material_property objects. Each Material_property defines the material_characteristics for exactly one Material.

4.3.144 Material to Specification

Each Material has the material_specification defined by zero, one, or many Specification objects. A Specification defines the material_specification for exactly one Material.

4.3.145 Material_property to Hardness

Each Material_property has the material_hardness defined by zero, one, or many Hardness objects. A Hardness defines material_hardness for exactly one Material_property.

4.3.146 Material_property to Property_parameter

Each Material_property has the property_characteristic defined by zero, one, or many Property_parameter objects. Each Property_parameters defines the property_characteristic for exactly one Material_property.

4.3.147 Mating_definition to Externally_defined_representation

Each Mating_definition has the standard_part_fastener defined by exactly zero, one, or many Externally_-definied_representation objects. Each Externally_definied_representation defines the standard_part_-fastener for zero, one, or many Mating_definition objects.

4.3.148 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.

4.3.149 Mating_definition to Shape_element

Each Mating_definition has the mating_shape defined by zero, two, or more Shape_element objects. Each Shape_element object defines the mating_shape for zero, one, or many Mating_definition objects.

4.3.150 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.

4.3.151 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 mating_part_definition for two, or more Mating_-definition_relationship objects.

4.3.152 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.

4.3.153 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.

4.3.154 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.

4.3.155 Multi_axis_feature to Planar_element

Each Multi_axis_feature has maximum_feature_limit defined by zero, one or many Planar_element. Each Planar_element defines the maximum_feature_limit for zero, one, or many Multi_axis_feature objects.

4.3.156 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.

4.3.157 Ngon_profile to Numeric_parameter

Each Ngon_profile 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_profile objects.

Each Ngon_profile has the diameter defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the diameter for zero, one, or many Ngon_profile objects.

Each Ngon_profile has the corner_radius defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the corner_radius for zero, one, or many Ngon_profile objects.

4.3.158 Numeric_parameter_with_tolerance to Limits_and_fits

Each Numeric_parameter_with_tolerance has the defined_value by zero or one Limit_and_fits. Each Limits_and_fits defines the defined_value for zero, one, or many Numeric_parameter_with_tolerance objects.

NOTE This assertion is established through Tolerance_definition_select.

4.3.159 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 Tolerance_definition_select.

4.3.160 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 Tolerance_definition_select.

4.3.161 Numeric_parameter_with_tolerance to Tolerance_range

Each Numeric_parameter_with_tolerance has the defined_value by zero or one tolerance_range. Each tolerance_range defines the defined_value for zero, one, or many Numeric_parameter_with_tolerance objects.

NOTE This assertion is established through Tolerance_definition_select.

4.3.162 Open_slot to Path

Each Open_slot has the course_of_travel defined by exactly one Path. Each Path defines the course_of_travel for zero, one, or many Open_slot objects.

4.3.163 Open_slot to Slot_end_type

Each Open_slot has the end_conditions defined by exactly two Slot_end_type objects. Each Slot_end_type defines the end_condition for zero, one, or many Open_slot objects.

4.3.164 Open_profile to Planar_element

Each Open_profile has the profile_limit defined by zero or one Planar_element. Each Planar_element defines the profile_limit for zero, one, or many Open_profile objects.

4.3.165 Outer_diameter to Numeric_parameter

Each Outer_diameter has the diameter defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the diameter for zero, one, or many Outer_diameter objects.

Each Outer_diameter has the feature_length defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the feature_length for zero, one, or many Outer_diameter objects.

4.3.166 Outer_diameter to Angle_taper

Each Outer_diameter has the reduced_size defined by zero or one Angle_taper. Each Angle_taper defines the reduced_size for zero, one, or many Outer_diameter objects.

NOTE This assertion is established through Taper_select.

4.3.167 Outer_diameter to Diameter_taper

Each Outer_diameter has the reduced_size defined by zero or one Diameter_taper. Each Diameter_taper defines the reduced_size for zero, one, or many Outer_diameter objects.

NOTE This assertion is established through Taper_select.

4.3.168 Outer_diameter to Directed_taper

Each Outer_diameter has the reduced_size defined by zero or one Directed_taper. Each Directed_taper defines the reduced_size for zero, one, or many Outer_diameter objects.

NOTE This assertion is established through Taper_select.

4.3.169 Outer_diameter_to_shoulder to Numeric_parameter

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

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

4.3.170 Outer_diameter_to_shoulder to Vee_profile

Each Outer_diamete_ to_shoulder has the v_shape_boundary defined by exactly one Vee_profile. Each Vee_profile defines the v_shape_boundary for zero, one, or many Outer_diameter_to_shoulder objects.

4.3.171 Parallelism_tolerance to Datum

Each Parallelism_tolerance has the geometric_reference defined by exactly one Datum. Each Datum defines the geometric_reference for zero, one, or many Parallelism_tolerance objects.

4.3.172 Parallelism_tolerance to Orientation

Each Parallelism_tolerance has an affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Parallelism_tolerance objects.

4.3.173 Part to Approval

Each Part has the manufacture_authorization defined by zero, one, or many Approval objects. Each Approval defines the manufacture_authorization for zero, one, or many Part objects.

4.3.174 Part to Ordered_part

Each Part has the quantity_ordered defined by one or more Ordered_part objects. Each Ordered_part defines quantity_ordered for exactly one Part.

4.3.175 Part to Organization

Each Part has the manufactured_by_organization defined by one or more Organization objects. Each Organization defines the manufactured_by_organization for zero, one or many Part objects.

Each Part has the owned_by_organization defined by one or more Organization objects. Each Organization defines the owned_by_organization for zero, one, or many Part objects.

4.3.176 Part to Person_in_organization

Each Part has the manufactured_by_person defined by one or more Person_in_organization objects. Each Person_in_organization defines the manufactured_by_person for zero, one, or many Part objects.

Each Part has the owned_by_person defined by one or more Person_in_organization objects. Each Person_in_organization defines the owned_by_person for zero, one, or many Part objects.

4.3.177 Part to Property

Each Part has the property_characteristics defined by zero, one, or many Property objects. Each Property defines the applied_to for exactly one Part.

4.3.178 Part to Shape

Each Part has the physical_form defined by exactly one Shape. Each Shape defines the physical_form for zero, one, or many Part objects.

4.3.179 Part_dimensioning_standard to Part

Each Part_dimensioning_standard has the applied_part defined by exactly one Part. Each Part defines the applied_part for zero, one, or many Part_dimensioning_standard objects.

4.3.180 Part_dimensioning_standard to Specification

Each Part_dimensioning_standard has the assigned_document defined by exactly one Specification objects. Each Specification defines the assigned_document for zero, one, or many Part_dimensioning_-standard objects.

4.3.181 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.

4.3.182 Part_placement to Shape

Each Part_placement has the oriented_physical_form defined by exactly one Shape. Each Shape defines the oriented_physical_form for zero, one, or many Part_placement objects.

4.3.183 Part_property to Property_parameter

Each Part_property has the property_characteristic defined by zero, one, or many Property_parameter objects. Each Property_parameter defines the property_characteristic for exactly one Part_property.

4.3.184 Partial_area_definition to Numeric_parameter

Each Partial_area_definition has the effective_length defined by exactly one Numeric_parameter. Each Numeric_parameter defines the effective_length for zero, one, or many Partial_area_definition objects.
Each Partial_area_definition has the maximum_length defined by zero or one Numeric_parameter. Each Numeric_parameter defines the maximum_length for zero, one, or many Partial_area_definition objects.

4.3.185 Partial_area_definition to Orientation

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

4.3.186 Partial_circular_path to Numeric_parameter

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

4.3.187 Partial_circular_profile to Numeric_parameter

Each Partial_circular_profile has the radius defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the radius for zero, one, or many Partial_circular_profile objects.

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

4.3.188 Partial_circular_shape_profile to Partial_circular_profile

Each Partial_circular_shape_profile has the open_boundary defined by exactly one Partial_-circular_profile. Each Partial_circular_profile defines the open_boundary for zero, one, or many Partial_-circular_shape_profile objects.

4.3.189 Path to Orientation

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

4.3.190 Perpendicularity_tolerance to Datum

Each Perpendicularity_tolerance has the geometric_reference defined by exactly one Datum. Each Datum defines the geometric_reference for zero, one, or many Perpendicularity_tolerance objects.

4.3.191 Perpendicularity_tolerance to Orientation

Each Perpendicularity_tolerance has the affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Perpendicularity_tolerance objects.

4.3.192 Person_in_organization to Organization

Each Person_in_organization has the company defined by exactly one Organization. Each Organization defines the company for zero, one, or many Person_in_organization objects.

4.3.193 Person_in_organization to Person

Each Person_in_organization has the employee defined by exactly one Person. Each Person defines the employee for zero, one, or many Person_in_organization objects.

4.3.194 Placed_target to Orientation

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

4.3.195 Planar_element to Direction_element

Each Planar_element has the normal defined by exactly one Direction_element. Each Direction_element defines the normal for zero, one, or many Planar_element objects.

4.3.196 Planar_element to Location_element

Each Planar_element has the location defined by exactly one Location_element. Each Location_element defines the location for zero, one, or many Planar_element objects.

4.3.197 Planar_face to Closed_profile

Each Planar_face has the face_boundary defined by zero or one Closed_profile object. Each Closed_profile defines the face_boundary for zero, one, or many Planar_face objects.

4.3.198 Planar_face to Direction_element

Each Planar_face has the removal_direction defined by exactly one Direction_element. Each Direction_-element defines the removal_direction for zero, one, or many Planar_face objects.

4.3.199 Planar_face to Linear_path

Each Planar_face has the course_of_travel defined by exactly one Linear_path. Each Linear_path defines the course_of_travel for zero, one, or many Planar_face objects.

4.3.200 Planar_face to Linear_profile

Each Planar_face has the removal_boundary defined by exactly one Linear_profile. Each Linear_profile defines the removal_boundary for zero, one, or many Planar_face objects.

4.3.201 Planar_face to Boss

Each Planar_face has the defined by zero, one or many Bose objects. Each Profile defines the for zero, one, or many Planar_face objects.

NOTE This assertion is established through Volume_select.

4.3.202 Planar_face to Numeric_parameter

Each Planar_face has the removal_depth defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the removal_depth for zero, one, or many Planar_face objects.

4.3.203 Planar_face to Protrusion

Each Planar_face has the volume_not_removed defined by zero, one or many Bose objects. Each Profile defines the volume_not_removed for zero, one, or many Planar_face objects.

NOTE This assertion is established through Volume_select.

4.3.204 Planar_pocket_bottom_condition to Direction_element

Each Planar_pocket_bottom_condition has the floor_normal defined by exactly one Direction_element. Each Direction_element defines floor_normal for zero, one, or many Planar_pocket_bottom_condition objects.

4.3.205 Planar_pocket_bottom_condition to Location_element

Each Planar_pocket_bottom_condition has the floor_location defined by exactly one Location_element. Each Location_element defines the floor_location for zero, one, or many Planar_pocket_bottom_-condition objects.

4.3.206 Planar_pocket_bottom_condition to Numeric_parameter

Each Planar_pocket_bottom_condition has the floor_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the floor_radius for zero, one, or many Planar_pocket_bottom_-condition objects.

4.3.207 Planar_profile_floor to Planar_element

Each Planar_profile_floor has the floor defined by exactly one Planar_element. Each Planar_element defines the floor for zero, one, or many Planar_profile_floor objects.

4.3.208 Planar_rib_top_floor to Closed_profile

Each Planar_rib_top_floor has the boundary defined by exactly one Closed_profile. Each Closed_profile defines the boundary for zero, one, or many Planar_rib_top_floor elements.

4.3.209 Planar_rib_top_floor to Planar_element

Each Planar_rib_top_floor has the floor_face defined by exactly one Planar_element. Each Planar_-element defines the floor_face for zero, one, or many Planar_rib_top_floor elements.

4.3.210 Planar_top_condition to Direction_element

Each Planar_top_condition has the top_normal defined by exactly one Direction_element. Each Direction_element defines the top_normal direction for zero, one, or many Planar_top_condition objects.

4.3.211 Planar_top_condition to Location_element

Each Planar_top_condition has the top_location defined by exactly one Location_element. Each Location_element defines the top_location for zero, one, or many Planar_top_condition objects.

4.3.212 Pocket to Angle_taper

Each Pocket has the change_in_boundary defined by zero, one, or many Angle_taper. Each Angle_taper defines the change_in_boundary for zero, one, or many Pocket objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.213 Pocket to Directed_taper

Each Pocket has the change_in_boundary defined by zero, one, or many Directed_taper. Each Directed_taper defines the change_in_boundary for zero, one, or many Pocket objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.214 Pocket to Linear_path

Each Pocket has the pocket_depth defined by exactly one Linear_path. Each Linear_path defines the pocket_depth for zero, one, or many Pocket objects.

4.3.215 Pocket to Numeric_parameter

Each Pocket has the base_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the base_radius for zero, one, or many Pocket objects.

4.3.216 Pocket to Pocket_bottom_condition

Each Pocket has the bottom_condition defined by exactly one Pocket_bottom_condition. Each Pocket_bottom_condition defines the bottom_condition for zero, one, or many Pocket objects.

4.3.217 Pocket to Through_pocket_bottom_condition

Each Pocket has the bottom_condition defined by exactly one Through_pocket_bottom_condition. Each Through_pocket_bottom_condition defines the bottom_condition for zero, one, or many Pocket objects.

4.3.218 Position_tolerance to Datum

Each Position_tolerance has the geometric_reference defined by one, two, or three Datum objects. Each Datum defines the geometric_reference for zero, one, or many Position_tolerance objects.

4.3.219 Position_tolerance to Orientation

Each Position_tolerance has the affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Position_tolerance objects.

4.3.220 Process_property to Property_parameter

Each Process_property has the property_characteristic defined by zero, one, or many Property_-parameters objects. Each Property_parameter defines the property characteristic for exactly one Process_property.

4.3.221 Profile to Orientation

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

4.3.222 Profile_feature to Linear_path

Each Profile_feature has the profile_swept_shape defined by exactly one Linear_path. Each Linear_path defines the profile_swept_shape for zero, one, or many Profile_feature objects.

4.3.223 Profile_floor to Numeric_parameter

Each Profile_floor has the floor_radius defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the floor_radius for zero, one, or many Planar_floor objects.

4.3.224 Project_order to Approval

Each Project_Order has the release_authorization defined by exactly one Approval. Each Approval defines the release_authorization for zero or one Project_Order.

4.3.225 Project_order to Digital_technical_data_package_work_order

Each Project_order has the technical_data_package_status defined by zero or one Digital_technical_-data_package_work_order objects. Each Digital_technical_data_package_work_order defines the technical_data_package_status for exactly one Project_order.

4.3.226 Project_order to Part

Each Project_order has the part_status defined by one or more Part objects. Each Part defines the part_status for exactly one Project_order.

4.3.227 Project_order to Pedigree_creation_order

Each Project_order has the pedigree_creation_status of zero or one Pedigree_creation_order. Each Pedigree_creation_order defines the pedigree_creation_status for exactly one Project_order.

4.3.228 Project_order to Requisition

Each Project_order has the ordered_resource defined by zero, one, or many Requisition objects. Each Requisition defines the ordered_resource for exactly one Project_order.

4.3.229 Project_order to Resource_acquisition_order

Each Project_order has the resource_acquisition_status defined by zero or one Resource_acquisition_-order. Each Resource_acquisition_order defines the resource_acquisition_order for exactly one Project_-order.

4.3.230 Project_order to Shop_work_order

Each Project_order has the shop_work_status defined by of zero or one Shop_work_order. Each Shop_-work_order defines the shop_work_status for exactly one Project_order.

4.3.231 Projection to Shape_element

Each Projection has the projection_end defined by exactly one Shape_element. Each Shape_element defines the projection_end for zero, one, or many Projection objects.

4.3.232 Property to Material_property

Each Property has the material_characteristic defined by zero, one, or many Material_property objects. Each Material_property defines the material_characteristic for one or more Property objects.

4.3.233 Property to Part_property

Each Property has the part_property_characteristic defined by zero, one, or many Part_property objects. Each Part_property defines the part_property_characteristic for one or more Property objects.

4.3.234 Property to Process_property

Each Property has the process_characteristic defined by zero, one, or many Process_property objects. Each Process_property defines the process_characteristic for one or more Property objects.

4.3.235 Property to Shape_aspect

Each Property has the property_characteristic defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the property_characteristic for zero, one, or many Property objects.

4.3.236 Property to Specification

Each Property has the property_description defined by zero, one, or many Specification objects. Each Specification defines the property_description for exactly one Property.

4.3.237 Property to Surface_property

Each Property has the surface_characteristic defined by zero, one, or many Surface_property objects. Each Surface_property defines the surface_characteristic for one or more Property objects.

4.3.238 Property_BSU to Class_BSU

Each Property_BSU has name_scope defined by exactly one Class_BSU. Each Class_BSU defines name_scope for zero, one, or many Property_BSU objects.

4.3.239 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.

4.3.240 Protrusion to Shape_element

Each Protrusion has the shape_volume defined by one or more Shape_element objects. Each Shape_element defines the shape_volume for zero, one, or many Protrusion objects.

4.3.241 Recess to Profile

Each Recess has the fillet_boundary defined by exactly one Profile. Each Profile defines the fillet_boundary for zero, one, or many Recess objects.

4.3.242 Recess to Pocket_bottom_condition

Each Recess has the bottom_condition defined by exactly one Pocket_bottom_condition. Each Pocket_-bottom_condition defines the bottom_condition for zero, one, or many Recess objects.

4.3.243 Recess to Boss

Each Recess has the volume_not_removed defined by zero, one or many Boss objects. Each Boss defines the volume_not_removed for zero, one, or many Recess objects.

NOTE This assertion is established through Volume_select.

4.3.244 Recess to Protrusion

Each Recess has the volume_not_removed defined by zero, one or many Protrusion objects. Each Protrusion defines the volume_not_removed for zero, one, or many Recess objects.

NOTE This assertion is established through Volume_select.

4.3.245 Rectangular_boss to Angle_taper

Each Rectangular_boss has the change_in_boundary defined by zero or one Angle_taper. Each Angle_-taper defines the change_in_boundary for zero, one, or many Rectangular_boss objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.246 Rectangular_boss to Directed_taper

Each Rectangular_boss t has the change_in_boundary defined by zero or one Directed_taper. Each Directed_taper defines the change_in_boundary for zero, one, or many Rectangular_boss objects.

NOTE This assertion is established through Angle_or_directed_taper_select.

4.3.247 Rectangular_boss to Rectangular_closed_profile

Each Rectangular_boss has the rectangular_profile defined by exactly one Rectangular_closed_profile. Each Rectangular_closed_profile defines the rectangular_profile for zero, one, or many Rectangular_boss objects.

4.3.248 Rectangular_closed_pocket to Rectangular_closed_profile

Each Rectangular_closed_pocket has a closed_boundary defined by exactly one Rectangular_closed_-profile. Each Rectangular_closed_profile identifies a closed_boundary for zero, one, or many Rectangular_closed_pocket objects.

4.3.249 Rectangular_closed_pocket to Boss

Each Rectangular_closed_pocket has the volume_not_removed defined by zero, one or many Bose objects. Each Boss defines the volume_not_removed for zero, one, or many Rectangular_closed_pocket objects.

NOTE This assertion is established through Volume_select.

4.3.250 Rectangular_closed_pocket to Protrusions

Each Rectangular_closed_pocket has the volume_not_removed defined by zero, one or many Protrusions objects. Each Protrusions defines the volume_not_removed for zero, one, or many Rectangular_closed_-pocket objects.

NOTE This assertion is established through Volume_select.

4.3.251 Rectangular_closed_profile to Numeric_parameter

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

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

Each Rectangular_closed_profile has the corner_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the corner_radius for zero, one, or many Rectangular_closed_profile objects.

4.3.252 Rectangular_closed_shape_profile to Rectangular_closed_profile

Each Rectangular_closed_shape_profile has the closed_boundary defined by exactly one Rectangular_-closed_profile. Each Rectangular_closed_profile defines the closed_boundary for zero, one, or many Rectangular_closed_shape_profile objects.

4.3.253 Rectangular_offset_pattern to Direction_element

Each Rectangular_offset_pattern has the offset_direction defined by exactly one Direction_element. Each Direction_element defines the offset_direction for zero, one, or many Rectangular_offset_pattern objects.

4.3.254 Rectangular_offset_pattern to Numeric_parameter

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

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

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

4.3.255 Rectangular_omit_pattern to Numeric_parameter

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

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

4.3.256 Rectangular_open_pocket to Boss

Each Rectangular_open_pocket has the volume_not_removed defined by zero, one or many Boss objects. Each Boss defines the volume_not_removed for zero, one, or many Rectangular_open_pocket objects

NOTE This assertion is established through Volume_select.

4.3.257 Rectangular_open_pocket to Protrusion

Each Rectangular_open_pocket has the volume_not_removed defined by zero, one or many Protrusions objects. Each Protrusions defines the volume_not_removed for zero, one, or many Rectangular_open_-pocket objects

NOTE This assertion is established through Volume_select.

4.3.258 Rectangular_open_pocket to Square_U_profile

Each Rectangular_open_pocket has the open_boundary defined by exactly one Square_U__profile. Each Square_U_profile defines the open_boundary for zero, one, or many Rectangular_open_pocket objects.

4.3.259 Rectangular_open_shape_profile to Square_U_profile

Each Rectangular_open_shape_profile has the open_boundary defined by exactly one Square_U_profile. Each Square_U_profile defines the open_boundary for zero, one, or many Rectangular_open_shape_-profile objects.

4.3.260 Rectangular_pattern to Direction_element

Each Rectangular_pattern has the row_layout_direction defined by exactly one Direction_element. Each Direction_element defines the row_layout_direction for zero, one, or many Rectangular_pattern objects.

Each Rectangular_pattern has the column_layout_direction defined by exactly one Direction_element. Each Direction_element defines the column_layout_direction for zero, one, or many Rectangular_pattern objects.

4.3.261 Rectangular_pattern to Numeric_parameter

Each Rectangular_pattern has the rows defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the rows for zero, one, or many Rectangular_pattern objects.

Each Rectangular_pattern has the columns defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the columns for zero, one, or many Rectangular_pattern objects.

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

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

4.3.262 Rectangular_pattern to Rectangular_offset_pattern

Each Rectangular_pattern has the relocated_base_feature defined by zero, one, or many Rectangular_-offset_pattern objects. Each Rectangular_offset_pattern defines the relocated_base_feature for one or more Rectangular_pattern objects.

4.3.263 Rectangular_pattern to Rectangular_omit_pattern

Each Rectangular_pattern has the missing_base_feature defined by zero, one, or many Rectangular_-omit_pattern objects. Each Rectangular_omit_pattern defines the missing_base_feature for one or more Rectangular_pattern objects.

4.3.264 Replicate_base to Machining_feature

Each Replicate_base has the base_feature defined by exactly one Machining_feature. Each Machining_-feature defines the base_feature for zero, one, or many Replicate_base objects.

NOTE This assertion is established through Replicate_base_select.

4.3.265 Replicate_base to Replicate_feature

Each Replicate_base has the base_feature defined by exactly one Replicate_feature. Each Replicate_-feature defines the base_feature for zero, one, many Replicate_base objects.

NOTE This assertion is established through Replicate_base_select.

4.3.266 Replicate_feature to Orientation

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

4.3.267 Replicate_feature to Replicate_base

Each Replicate_feature has the replicate_base_feature defined by exactly one Replicate_base. Each Replicate_base defines the replicate_base_feature for one or more Replicate_feature objects.

4.3.268 Revolved_feature to Numeric_parameter

Each Revolved_feature has the radius defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the radius for zero, one, or many Revolved_feature objects.

4.3.269 Revolved_feature to Direction_element

Each Revolved_feature has the material_side defined by exactly one Direction_element. Each Direction_element defines the material_side for zero, one, or many Revolved_feature objects.

4.3.270 Revolved_flat to Linear_profile

Each Revolved_flat has the flat_edge_shape defined by exactly one Linear_profile. Each Linear_profile defines the flat_edge_shape for zero, one, or many Revolved_flat objects.

4.3.271 Revolved_round to Partial_circular_profile

Each Revolved_round has the rounded_edge_shape defined by exactly one Partial_circular_profile. Each Partial_circular_profile defines the rounded_edge_shape for zero, one, or many Revolved_round objects.

4.3.272 Rib_top to Direction_element

Each Rib_top has the removal_direction defined by exactly one Direction_element. Each Direction_-element defines the removal_direction for zero, one or many Rib_top objects.

4.3.273 Rib_top to Rib_top_floor

Each Rib_top has the floor_condition defined by exactly one Rib_top_floor. Each Rib_top_floor defines the floor_condition for zero, one, or many Rib_top objects.

4.3.274 Round_hole to Blind_bottom_condition

Each Round_hole has the bottom_condition defined by exactly one Blind_bottom_condition. Each Blind_bottom_condition defines the bottom_condition for zero, one, or many Round_hole objects.

NOTE This assertion is established through Hole_bottom_condition_select.

4.3.275 Round_hole to Through_bottom_condition

Each Round_hole has the bottom_condition defined by exactly one Through_bottom_condition. Each Through_bottom_condition defines the bottom_condition for zero, one, or many Round_hole objects.

NOTE This assertion is established through Hole_bottom_condition_select.

4.3.276 Round_hole to Circular_closed_profile

Each Round_hole has the diameter defined by exactly one Circular_closed_profile. Each Circular_-closed_profile defines the diameter for zero, one, or many Round_hole objects.

4.3.277 Round_hole to Linear_path

Each Round_hole has the hole_depth defined by exactly one Linear_path. Each Linear_path defines the hole_depth for zero, one, or many Round_hole objects.

4.3.278 Round_hole to Angle_taper

Each Round_hole has the change_in_diameter defined by zero or one Angle_taper. Each Angle_taper defines change_in_diameter for zero, one, or many Round_hole objects.

NOTE This assertion is established through Taper_select.

4.3.279 Round_hole to Diameter_taper

Each Round_hole has the change_in_diameter defined by zero or one Diameter_taper. Each Diameter_-taper defines change_in_diameter for zero, one, or many Round_hole objects.

NOTE This assertion is established through Taper_select.

4.3.280 Round_hole to Directed_taper

Each Round_hole has the change_in_directed defined by zero or one Diameter_taper. Each Directed_-taper defines change_in_diameter for zero, one, or many Round_hole objects.

NOTE This assertion is established through Taper_select.

4.3.281 Rounded_end to Linear_path

Each Rounded_end has the course_of_travel defined by exactly one Linear_path. Each Linear_path defines the course_of_travel for zero, one, or many Rounded_end objects.

4.3.282 Rounded_end to Partial_circular_profile

Each Rounded_end has the partial_circular_boundary defined by exactly one Partial_circular_profile. Each Partial_circular_profile defines a partial_circular_boundary for zero, one, or many Rounded_end objects.

4.3.283 Rounded_U_profile to Numeric_parameter

Each Rounded_U_profile has the depth defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the depth for zero, one, or many Rounded_U_profile objects.

Each Rounded_U_profile has the width defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the width for zero, one, or many Rounded_U_profile objects.

4.3.284 Second_chamfer_offset to Face_shape_element

Each Second_chamfer_offset has the second_face defined by exactly one Face_shape_element. Each Face_shape_element defines the second_face for zero, one, or many Second_chamfer_offset objects.

4.3.285 Second_offset to Numeric_parameter

Each Second_offset has the offset_amount defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the offset_amount for zero, one, or many Second_offset objects.

4.3.286 Shape to Base_shape

Each Shape has the base_shape_definition defined by zero or one Base_shape. Each Base_shape defines the base_shape_definition for exactly one Shape.

4.3.287 Shape to B-rep_shape_representation

Each Shape has the B-rep_form defined by zero, one, or many B-rep_shape_representation objects. Each B-rep_shape_representation defines the B-rep_form for exactly one Shape.

4.3.288 Shape to Shape_aspect

Each Shape has the element defined by zero, one, or many Shape_aspect objects. Each Shape_aspect defines the element for zero, one, or many Shape objects.

4.3.289 Shape_aspect to B-rep_model_element

Each Shape_aspect has the B-rep_shape defined by zero, one, or many B-rep_model_element objects. Each B-rep_model_element defines the B-rep_shape for exactly one Shape_aspect.

4.3.290 Shape_aspect to B-rep_shape_aspect_representation

Each Shape_aspect has the B-rep_form defined by zero, one, or many B-rep_shape_aspect_representation objects. Each B-rep_shape_aspect_representation defines the B-rep_form for exactly one Shape_aspect.

4.3.291 Shape_aspect to Shape_element

Each Shape_aspect has the element defined by zero or one Shape_element. Each Shape_element defines the element for exactly one Shape_aspect.

4.3.292 Shape_profile to Direction_element

Each Shape_profile has the removal_direction defined by exactly one Direction_element. Each Direction_element defines the removal_direction for zero, one or many Shape_profile objects.

4.3.293 Shape_profile to Profile_floor

Each Shape_profile has the floor_condition defined by exactly one Profile_floor. Each Profile_floor defines the floor_condition for zero, one, or many Shape_profile objects.

NOTE This assertion is established through Profile_select.

4.3.294 Shape_profile to Through_profile_floor

Each Shape_profile has the floor_condition defined by exactly one Profile_floor. Each Profile_floor defines the floor_condition for zero, one, or many Shape_profile objects.

NOTE This assertion is established through Profile_select.

4.3.295 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 alternate_material_definition for one or more Single_piece_part.

4.3.296 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.

4.3.297 Size_tolerance to Shape_element

Each Size_tolerance has the applied_shape defined exactly one Shape_element. Each Shape_element defines the applied_shape for zero, one, or many Size_dimension objects.

4.3.298 Slot to Open_profile

Each Slot has the sweep_shape defined by exactly one Open_profile. Each Open_profile defines the sweep_shape for zero, one, or many Slot objects.

4.3.299 Specification to Specification_usage_constraint

Each Specification has the constraint defined by zero, one, or many Specification_usage_constraint objects. Each Specification_usage_constraint defines the constraint for exactly one Specification.

4.3.300 Spherical_cap to Numeric_parameter

Each Spherical_cap has the internal_angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the internal_angle for zero, one, or many Spherical_cap objects.

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

4.3.301 Spherical_hole_bottom to Numeric_parameter er

Each Spherical_hole_bottom has the radius defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the radius for zero, one, or many Spherical_hole_bottom objects.

4.3.302 Square_U_profile to Numeric_parameter

Each Square_U_profile has the depth defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the depth for zero, one, or many Square_U_profile objects.

Each Square_U_profile has the first_angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the first_angle for zero, one, or many Square_U_profile objects.

Each Square_U_profile has the first_radius defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the first_radius for zero, one, or many Square_U_profile objects.

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

Each Square_U_profile has the second_radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the second_radius for zero, one, or many Square_U_profile objects.

Each Square_U_profile has the width defined by exactly one Numeric_parameter Each Numeric_-parameter defines the width for zero, one, or many Square_U_profile objects.

4.3.303 Step to Vee_profile

Each Step has the removal_boundary defined by exactly one Vee_profile. Each Vee_profile defines the removal_boundary for zero, one, or many Step objects.

4.3.304 Step to Linear_path

Each Step has the course_of_travel defined by exactly one Linear_path. Each Linear_path defines the course_of_travel for zero, one, or many Step objects.

4.3.305 Step to Boss

Each Step has the volume_not_removed defined by zero, one or many Boss objects. Each Boss defines the volume_not_removed for zero, one, or many Step objects.

NOTE This assertion is established through Volume_select.

4.3.306 Step to Protrusion

Each Step has the volume_not_removed defined by zero, one or many Protrusions objects. Each Protrusions defines the volume_not_removed for zero, one, or many Step objects.

NOTE This assertion is established through Volume_select.

4.3.307 Straightness_tolerance to Orientation

Each Straightness_tolerance has the affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Straightness_tolerance objects.

4.3.308 Surface_profile_tolerance to Datum

Each Surface_profile has the geometric_reference defined by zero, one, two, or three Datum objects. Each Datum defines the geometric_reference for zero, one, or many Surface_profile objects.

4.3.309 Surface_property to Property_parameter

Each Surface_property has the property_characteristic defined by zero, one, or many Property_parameter objects. A Property_parameter defines the property_characteristic for exactly one Surface_property.

4.3.310 Symmetry_tolerance to Datum

Each Symmetry_tolerance has the geometric_reference defined by one, two, or three Datum objects. Each Datum defines the geometric_reference for zero, one, or many Symmetry_tolerance objects.

4.3.311 Symmetry_tolerance to Orientation

Each Symmetry_tolerance has the affected_plane defined by zero or one Orientation. Each Orientation defines the affected_plane for zero, one, or many Symmetry_tolerance objects.

4.3.312 Target_area to Shape_element

Each Target_area has the area_shape defined by exactly one Shape_element. Each Shape_element defines the area_shape by zero, one, or many Shape_element objects.

4.3.313 Tee_profile to Numeric_parameter

Each Tee_profile has the cross_bar_depth defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the cross_bar_depth for zero, one, or many Tee_profile objects.

Each Tee_profile has the cross_bar_width defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the cross_bar_width for zero, one, or many Tee_profile objects.

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

Each Tee_profile has the first angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the first angle for zero, one, or many Tee_profile objects.

Each Tee_profile has the first offset defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the first offset for zero, one, or many Tee_profile objects.

Each Tee_profile has the second angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the second angle for zero, one, or many Tee_profile objects.

Each Tee_profile has the second offset defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the second offset for zero, one, or many Tee_profile objects.

Each Tee_profile has the radius defined by zero or one Numeric_parameter. Each Numeric_parameter defines the radius for zero, one, or many Tee_profile objects.

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

4.3.314 Thread to Descriptive_parameter

Each Thread has the fit_class defined by exactly one Descriptive_parameter. Each Descriptive_parameter defines the fit_class for zero, one, or many Thread objects.

Each Thread has the fit_class_2 defined by exactly one Descriptive_parameter. Each Descriptive_-parameter defines the fit_class_2 for zero, one, or many Thread objects.

Each Thread has the form defined by exactly one Descriptive_parameter. Each Descriptive_parameter defines the form for zero, one, or many Thread objects.

Each Thread has the qualifier defined by zero or one Descriptive_parameter. Each Descriptive_parameter defines the qualifier for zero, one, or many Thread objects.

Each Thread has the thread_hand defined by exactly one Descriptive_parameter. Each Descriptive_-parameter defines the thread_hand for zero, one, or many Thread objects.

4.3.315 Thread to Numeric_parameter

Each Thread has the nominal_size defined by zero or one Numeric_parameter. Each Numeric_parameter defines the nominal_size for zero, one, or many Thread objects.

Each Thread has the number_of_threads defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the number_of_threads for zero, one, or many Thread objects.

4.3.316 Thread to Partial_area_definition

Each Thread has the partial_profile defined by exactly one Partial_area_definition. Each Partial_area_-definition defines the partial_profile for zero, one, or many Thread objects.

4.3.317 Thread to Shape_element

Each Thread has the applied_shape defined by exactly one Shape_element. Each Shape_element defines the applied_shape for zero, one, or many Thread objects.

4.3.318 Thread to Thread_runout

Each Thread has the runout defined by zero or one Thread_runout. Each Thread_runout defines the runout for zero, one, or many Thread objects.

4.3.319 Thread_runout to Numeric_parameter

Each Thread_runout has the length_of_runout defined by zero or one Numeric_parameter. Each Numeric_parameter defines the length_of_runout for zero, one, or many Thread_runout objects.

4.3.320 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.

4.3.321 Tolerance_value to Limits_and_fits

Each Tolerance_value has the defined_value by zero or 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.

4.3.322 Tolerance_value to Plus_minus_value

Each Tolerance_value has the defined_value by zero or one Plus_minus_value. Each Plus_minus_value defines the defined_values for zero, one, or many Tolerance_value objects.

NOTE This assertion is established through Tolerance_definition_select.

4.3.323 Tolerance_value to Tolerance_limit

Each Tolerance_value has the defined_value by zero or one tolerance_limit. Each tolerance_limit defines the defined_value for zero, one, or many Tolerance_value objects.

NOTE This assertion is established through Tolerance_definition_select.

4.3.324 Tolerance_value to Tolerance_range

Each Tolerance_value has the defined_value by zero or one tolerance_range. Each tolerance_range defines the defined_value for zero, one, or many Tolerance_value objects.

NOTE This assertion is established through Tolerance_definition_select.

4.3.325 Tolerance_zone to Tolerance_zone_definition

Each Tolerance_zone has zone_definition defined by exactly one Tolerance_zone_definition. Each Tolerance_zone_definition defines zone_definition for zero, one, or many Tolerance_zone objects.

4.3.326 Tolerance_zone to Projection

Each Tolerance_zone has extended_shape defined by zero, one, two, or three Projection objects. Each Projection defines extended_shape for zero, one, or many Tolerance_zone objects.

4.3.327 Tolerance_zone_definition to Shape_element

Each Tolerance_zone_definition has the first_element defined by exactly one Shape_element. Each Shape_element defines the first_element for zero, one, or many Tolerance_zone_definition objects.

Each Tolerance_zone_definition has the second_element defined by zero or one Shape_element. Each Shape_element defines the second_element for zero, one, or many Tolerance_zone_definition objects.

4.3.328 Total_runout_tolerance to Datum

Each Total_runout_tolerance has the geometric_reference defined by one or two Datum objects. Each Datum defines the geometric_reference for zero, one, or many Total_runout_tolerance objects.

4.3.329 Turned_knurl to Numeric_parameter

Each Turned_knurl has the tooth_depth defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the tooth_depth for zero, one, or many Turned_knurl objects.

Each Turned_knurl has the root_fillet defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the root_fillet for zero, one, or many Turned_knurl objects.

Each Turned_knurl has the major_diameter defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the major_diameter for zero, one, or many Turned_knurl objects.

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

Each Turned_knurl has the number_of_teeth defined by zero or one Numeric_parameter. Each Numeric_parameter defines the number_of_teeth for zero, one, or many Turned_knurl objects.

Each Turned_knurl has the diametral_pitch defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the diametral_pitch for zero, one, or many Turned_knurl objects.

4.3.330 Vee_profile to Numeric_parameter

Each Vee_profile has the first_length defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the first_length for zero, one, or many Vee_profile objects.

Each Vee_profile has the second_length defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the second_length for zero, one, or many Vee_profile objects.

Each Vee_profile has the profile_radius defined by zero or one Numeric_parameter. Each Numeric_-parameter defines the profile_radius for zero, one, or many Vee_profile objects.

Each Vee_profile has the profile_angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the profile_angle for zero, one, or many Vee_profile objects.

Each Vee_profile has the tilt_angle defined by exactly one Numeric_parameter. Each Numeric_-parameter defines the tilt_angle for zero, one, or many Vee_profile objects.

4.3.331 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.

4.3.332 Woodruff_slot_end_type to Numeric_parameter

Each Woodruff_slot_end_type has the radius defined by exactly one Numeric_parameter. Each Numeric_parameter defines the radius for zero, one, or many Woodruff_slot_end_type 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 design_exception UoF

5.1.1.1 Design_exception_notice

AIM element: versioned_action_request
Source: ISO 10303-41
Rules: dependent_instantiable_action_request_status – (See 5.2.4.6)

5.1.1.1.1 issuing_date

AIM element: date
Source: ISO 10303-41
Rules:
Reference Path: versioned_action_request
feature_based_pp_dated_item = versioned_action_request
feature_based_pp_dated_item <-
feature_based_pp_date_assignment.items[i]
feature_based_pp_date_assignment <=
{date_assignment
date_assignment.role ->
date_role
date_role.name = 'issuing date'}
date_assignment
date_assignment.assigned_date ->
date

5.1.1.1.2 notice_description

AIM element: versioned_action_request.purpose
Source: ISO 10303-41

5.1.1.1.3 notice_number

AIM element: versioned_action_request.id
Source: ISO 10303-41

5.1.1.1.4 technical_recommendation

AIM element: action_method
Reference Path: versioned_action_request <-
action_request_solution.request
action_request_solution
action_request_solution.method ->
action_method

5.1.1.1.5 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
feature_based_pp_document_item = action_method
feature_based_pp_document_item <-
feature_based_pp_document_reference.items[i]
feature_based_pp_document_reference <=
document_reference
document_reference.assigned_document ->
document
{document
document.kind ->
document_type
document_type.product_data_type = 'engineering change proposal'}

5.1.1.1.6 design_exception_notice to part (as discrepant_part)

AIM element: PATH
Reference Path: versioned_action_request <-
action_request_assignment.assigned_action_request
action_request_assignment =>
feature_based_pp_action_request_assignment
feature_based_pp_action_request_assignment.items[i] ->
feature_based_pp_action_request_item
feature_based_pp_action_request_item = product_definition_formation
product_definition_formation

5.1.1.2 Engineering_change_order

AIM element: action_directive
Source: ISO 10303-41

5.1.1.2.1 change_order_number

AIM element: action_directive.name
Source: ISO 10303-41

5.1.1.2.2 engineering_change_order to part (as new_version)

AIM element: PATH
Reference Path: action_directive <-
directed_action.directive
directed_action <=
executed_action <=
action <-
action_assignment.assigned_action
action_assignment =>
feature_based_pp_action_assignment
feature_based_pp_action_assignment.items[i] ->
feature_based_pp_action_item
feature_based_pp_action_item = product_definition_formation
product_definition_formation

5.1.1.3 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'

5.1.1.3.1 change_proposal_number

AIM element: document.id
Source: ISO 10303-41

5.1.1.3.2 engineering_change_proposal to engineering_change_order (as incorporated_proposal)

AIM element: PATH
Reference Path: document<-
document_reference.assigned_document
document_reference=>
feature_based_pp_document_reference
feature_based_pp_document_reference.items[i]->
feature_based_pp_document_item
feature_based_pp_document_item = action_method
action_method<-
action.chosen_method
action=>
executed_action=>
directed_action
directed_action.directive->
action_directive

5.1.2 feature_definition_item UoF

5.1.2.1 Angle_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: taper <=
shape_aspect
{shape_aspect
shape_aspect.description = 'angle taper'}
{shape_aspect
shape_aspect.of_shape->
product_definition_shape<=
property_definition
property_definition.definition->
characterized_definition=characterized_object
characterized_object=>
feature_component_definition}

5.1.2.1.1 angle_taper to numeric_parameter (as angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: taper <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'taper angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.2 Blind_bottom_condition

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
(shape_aspect.description = 'conical')
(shape_aspect.description = 'flat')
(shape_aspect.description = 'flat with radius')
(shape_aspect.description = 'flat with taper')
(shape_aspect.description = 'spherical')}
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.2.1 start_or_end

AIM element: shape_aspect_relationship.name
Source: ISO 10303-41
Reference Path: round_hole <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship
shape_aspect_relationship.name = 'hole depth start'
shape_aspect_relationship.name = 'hole depth end'

5.1.2.3 Boss_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss_top <=
shape_aspect
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.3.1 start_or_end

AIM element: shape_aspect_relationship.name
Source: ISO 10303-41
Reference Path: boss_top <=
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.name
{(shape_aspect_relationship.name = 'boss height start')
(shape_aspect_relationship.name = 'boss height end')}

5.1.2.3.2 boss_top to numeric_parameter (as top_radius)

AIM element: PATH
Source: ISO 10303-41
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'top radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.4 Chamfer_angle

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.2.4.1 chamfer_angle to numeric_parameter (as angle_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'offset angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.5 Circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
(shape_aspect.description = 'partial circular')
(shape_aspect.description = 'complete circular')}

5.1.2.5.1 circular_path to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.6 Complete_circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complete circular'}

5.1.2.7 Conical_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'conical'}

5.1.2.7.1 conical_hole_bottom to numeric_parameter (as tip_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tip angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.7.2 conical_hole_bottom to numeric_parameter (as tip_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tip radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.8 Diameter_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: taper <=
shape_aspect
{shape_aspect
shape_aspect.description = 'diameter taper'}
{shape_aspect
shape_aspect.of_shape->
product_definition_shape<=
property_definition
property_definition.definition->
characterized_definition=characterized_object
characterized_object=>
feature_component_definition}

5.1.2.8.1 diameter_taper to numeric_parameter (as final_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: taper <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'final diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.9 Directed_taper

AIM element: taper
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: taper <=
shape_aspect
{shape_aspect
shape_aspect.description = 'directed taper'} {shape_aspect
shape_aspect.of_shape->
product_definition_shape<=
property_definition
property_definition.definition->
characterized_definition=characterized_object
characterized_object=>
feature_component_definition}

5.1.2.9.1 directed_taper to direction_element (as direction)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: taper <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.2.9.2 directed_taper to numeric_parameter (as angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: taper <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.10 First_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'first offset'}

5.1.2.10.1 first_offset to face_shape_element (as face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'first face shape'}
representation =>
shape_representation=>
face_shape_representation

5.1.2.10.2 first_offset to numeric_parameter (as offset_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'offset amount'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.11 Flat_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat'}

5.1.2.12 Flat_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat'}

5.1.2.12.1 flat_slot_end_type to numeric_parameter (as first_radius)

AIM element: PATH

Rules:
dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)

Reference Path:
```
slot_end <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}
```

5.1.2.12.2 flat_slot_end_type to numeric_parameter (as second_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: slot_end <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.13 Flat_with_radius_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat with radius'}

5.1.2.13.1 flat_with_radius_hole_bottom to numeric_parameter (as corner_-radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'corner radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.14 Flat_with_taper_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'flat with taper'}

5.1.2.14.1 flat_with_taper_hole_bottom to numeric_parameter (as final_-diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'final diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.14.2 flat_with_taper_hole_bottom to numeric_parameter (as taper angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'taper angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.15 General_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.2.15.1 general_path to path_element (as sweep_path)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'sweep path'}
representation =>
shape_representation=>
path_shape_representation

5.1.2.16 General_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: profile_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.2.16.1 general_profile_floor to face_shape_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'floor'}
representation =>
shape_representation=>
face_shape_representation

5.1.2.17 General_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.2.17.1 general_pocket_bottom_condition to face_shape_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'floor face'}
representation =>
shape_representation=>
face_shape_representation

5.1.2.17.2 general_pocket_bottom_condition to numeric_parameter (as floor_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.18 General_rib_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rib_top_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.2.18.1 general_rib_top_floor to ordered_face_element (as rib_top_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: rib_top_floor<=
shape_aspect
shape_definition=shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'rib top face'}
representation =>
shape_representation =>
face_shape_representation

5.1.2.19 General_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss_top <=
shape_aspect
{shape_aspect
shape_aspect.description = 'complex'}

5.1.2.19.1 general_top_condition to face_shape_element (as top_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
face_shape_representation

5.1.2.20 Linear_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'linear'}

5.1.2.20.1 linear_path to direction_element(as direction)

AIM element: PATH
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
direction_shape_representation

5.1.2.20.2 linear_path to numeric_parameter (as distance)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'distance'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.21 Loop_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'loop'}

5.1.2.22 Open_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'open'}

5.1.2.23 Partial_area_definition

AIM element: applied_area
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: applied_area <=
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape}

5.1.2.23.1 partial_area_definition to numeric_parameter (as effective_length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: applied_area <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'effective length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.23.2 partial_area_definition to numeric_parameter (as maximum_-length)

AIM element: PATH

Rules:
dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)

Reference Path:
applied_area <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'maximum length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.23.3 partial_area_definition to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: applied_area <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.2.24 Partial_circular_path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.description = 'partial circular'}

5.1.2.24.1 partial_circular_path to numeric_parameter (as sweep_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'sweep angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.2.25 Path

AIM element: path_feature_component
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: path_feature_component <=
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape<=
property_definition
property_definition.definition->
characterized_definition=characterized_object
characterized_object=>
feature_component_definition}

5.1.2.25.1 path to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: path_feature_component <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.2.26 Planar_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.2.26.1 planar_pocket_bottom_condition to direction_element (as floor_-normal)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path:
```
pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'floor normal'}
representation =>
shape_representation=>
direction_shape_representation
```

5.1.2.26.2 planar_pocket_bottom_condition to location_element (as floor_-location)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path:
```
pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'floor location'}
representation =>
shape_representation=>
location_shape_representation
```

5.1.2.26.3 planar_pocket_bottom_condition to numeric_parameter (as floor_-radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: pocket_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.27 Planar_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: profile_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.2.27.1 planar_profile_floor to planar_element (as floor)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'floor'}
representation =>
planar_shape_representation

5.1.2.28 Planar_rip_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rib_top_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.2.28.1 planar_rib_top_floor to closed_profile (as boundary)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rib_top_floor <=
{shape_aspect
shape_aspect.description = 'boundary occurrence'}
shape_aspect<-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description= 'rib top floor boundary'}
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)

5.1.2.28.2 planar_rib_top_floor to planar_element (as planar_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: rib_top_floor<=
shape_aspect
shape_definition=shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'rib top face'}
representation =>
shape_representation =>
planar_shape_representation

5.1.2.28.3 Planar_top_condition

AIM element: boss_top
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss_top <=
shape_aspect
{shape_aspect
shape_aspect.description = 'planar'}

5.1.2.28.4 planar_top_condition to direction_element (as top_normal)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
direction_shape_representation

5.1.2.28.5 planar_top_condition to location_element (as top_location)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: boss_top <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
location_shape_representation

5.1.2.29 Pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.29.1 start_or_end

AIM element: shape_aspect_relationship.name
Source: ISO 10303-41
Reference Path: pocket_bottom <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship
shape_aspect_relationship.name = 'pocket depth start'
shape_aspect_relationship.name = 'pocket depth end'

5.1.2.30 Profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: profile_floor<=
shape_aspect
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.30.1 start_or_end

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'shape profile floor orientation'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'shape profile start')
(descriptive_representation_item.description = 'shape profile end')}

5.1.2.30.2 profile_floor to numeric_parameter (as floor radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: profile_floor <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.31 Radiused_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'radiused'}

5.1.2.32 Rip_top_floor

AIM element: rib_top_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rib_top_floor <=
shape_aspect
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.33 Second_chamfer_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.2.33.1 second_chamfer_offset to face_shape_element (as second_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'second face shape'}
representation =>
shape_representation=>
face_shape_representation

5.1.2.34 Second_offset

AIM element: chamfer_offset
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer_offset <=
shape_aspect
{shape_aspect
shape_aspect.description = 'second offset'}

5.1.2.34.1 second_offset to numeric_parameter (as offset_amount)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: chamfer_offset <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'offset amount'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.35 Slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.35.1 first_or_second

AIM element: shape_aspect_relationship.name
Source: ISO 10303-41
Reference Path: slot_end <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.name
{(shape_aspect_relationship.name = 'course of travel start')
(shape_aspect_relationship.name = 'course of travel end')}

5.1.2.36 Spherical_hole_bottom

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'spherical'}

5.1.2.36.1 spherical_hole_bottom to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: hole_bottom <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.2.37 Through_bottom_condition

AIM element: hole_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: hole_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'through'}
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.38 Through_pocket_bottom_condition

AIM element: pocket_bottom
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket_bottom <=
shape_aspect
{shape_aspect
shape_aspect.description = 'through'}
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.39 Through_profile_floor

AIM element: profile_floor
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: profile_floor <=
shape_aspect
{shape_aspect
shape_aspect.description = 'through'}
{shape_aspect.of_shape ->
product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition
characterized_definition = characterized_object
characterized_object =>
feature_component_definition}

5.1.2.40 Woodruff_slot_end_type

AIM element: slot_end
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot_end <=
shape_aspect
{shape_aspect
shape_aspect.description = 'woodruff'}

5.1.2.40.1 woodruff_slot_end_type to nu.meric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: slot_end <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
.{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3 feature_profile UoF

5.1.3.1 Circular_closed_profile

AIM element: circular_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: circular_closed_profile <=
shape_aspect

5.1.3.1.1 circular_closed_profile to numeric_parameter (as diameter)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: circular_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.2 Closed_profile

AIM element: (circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (circular_closed_profile <=)
(closed_path_profile <=)
(ngon_closed_profile <=)
(rectangular_closed_profile <=)
shape_aspect

5.1.3.3 General_closed_profile

AIM element: closed_path_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: closed_path_profile <=
shape_aspect

5.1.3.3.1 general_closed_profile to path_element (as closed_profile_shape)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: closed_path_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
path_shape_representation

5.1.3.4 General_open_profile

AIM element: open_path_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: open_path_profile <=
shape_aspect

5.1.3.4.1 general_open_profile to path_element (as enclosed_boundary)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: open_path_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation=>
path_shape_representation

5.1.3.5 Linear_profile

AIM element: linear_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: linear_profile <=
shape_aspect

5.1.3.5.1 linear_profile to numeric_parameter (as profile_length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: linear_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'profile length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.6 Ngon_profile

AIM element: ngon_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: ngon_closed_profile <=
shape_aspect

5.1.3.6.1 circumscribed_or_across_flats

AIM element: (representation_item.name = 'circumscribed diameter')
(representation_item.name =
'diameter across flats')
Source: ISO 10303-522
Reference Path: ngon_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
representation_item
(representation_item.name = 'circumscribed diameter')
(representation_item.name = 'diameter across flats')

5.1.3.6.2 ngon_profile to numeric_parameter (as diameter)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: ngon_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
(representation_item.name = 'circumscribed diameter')
(representation_item.name = 'diameter across flats')}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.6.3 ngon_profile to numeric_parameter (as corner_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: ngon_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'corner radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.6.4 ngon_profile to numeric_parameter (as number_of_sides)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: ngon_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = '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}

5.1.3.7 Open_profile

AIM element: (linear_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (linear_profile <=)
(open_path_profile <=)
(partial_circular_profile <=)
(rounded_u_profile <=)
(square_u_profile <=)
(tee_profile <=)
(vee_profile <=)
shape_aspect

5.1.3.7.1 open_profile to planar_element (as profile_limit)

AIM element: PATH
Source: ISO 10303-522
Reference Path: (rounded_u_profile <=)
(square_u_profile <=)
(open_path_profile <=)
(tee_profile <=)
(vee_profile <=)
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
representation=>
{representation.name='profile limit'}
shape_representation=>
planar_shape_representation

5.1.3.8 Partial_circular_profile

AIM element: partial_circular_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: partial_circular_profile <=
shape_aspect

5.1.3.8.1 partial_circular_profile to numeric_parameter (as radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: partial_circular_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.8.2 partial_circular_profile to numeric_parameter (as sweep_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: partial_circular_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'sweep angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.9 Profile

AIM element: (circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(linear_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (circular_closed_profile <=)
(closed_path_profile <=)
(ngon_closed_profile <=)
(rectangular_closed_profile <=)
(linear_profile <=)
(open_path_profile <=)
(partial_circular_profile <=)
(rounded_u_profile <=)
(square_u_profile <=)
(tee_profile <=)
(vee_profile <=)
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape<=
property_definition
property_definition.definition->
characterized_definition=characterized_object
characterized_object=>
feature_component_definition}

5.1.3.9.1 profile to orientation (as placement)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (circular_closed_profile <=)
(closed_path_profile <=)
(ngon_closed_profile <=)
(rectangular_closed_profile <=)
(linear_profile <=)
(open_path_profile <=)
(partial_circular_profile <=)
(rounded_u_profile <=)
(square_u_profile <=)
(tee_profile <=)
(vee_profile <=)
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.3.10 Rectangular_closed_profile

AIM element: rectangular_closed_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rectangular_closed_profile <=
shape_aspect

5.1.3.10.1 rectangular_closed_profile to numeric_parameter (as corner_-radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'corner radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.10.2 rectangular_closed_profile to numeric_parameter (as profile_-length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.10.3 rectangular_closed_profile to numeric_parameter (as profile_-width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_closed_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.11 Rounded_u_profile

AIM element: rounded_u_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rounded_u_profile <=
shape_aspect

5.1.3.11.1 rounded_u_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rounded_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.11.2 rounded_u_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rounded_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.12 Square_u_profile

AIM element: square_u_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: square_u_profile <=
shape_aspect

5.1.3.12.1 square_u_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:
square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.12.2 square_u_profile to numeric_parameter (as first_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.12.3 square_u_profile to numeric_parameter (as first_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.12.4 square_u_profile to numeric_parameter (as second_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.12.5 square_u_profile to numeric_parameter (as second_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.12.6 square_u_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: square_u_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13 Tee_profile

AIM element: tee_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: tee_profile <=
shape_aspect

5.1.3.13.1 tee_profile to numeric_parameter (as cross_bar_depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'cross bar depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.2 tee_profile to numeric_parameter (as cross_bar_width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'cross bar width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.3 tee_profile to numeric_parameter (as depth)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.4 tee_profile to numeric_parameter (as first_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.13.5 tee_profile to numeric_parameter (as first_offset)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.6 tee_profile to numeric_parameter (as radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.7 tee_profile to numeric_parameter (as second_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.13.8 tee_profile to numeric_parameter (as second_offset)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.13.9 tee_profile to numeric_parameter (as width)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: tee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.14 Vee_profile

AIM element: vee_profile
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: vee_profile <=
shape_aspect

5.1.3.14.1 vee_profile to numeric_parameter (as first length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: vee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.14.2 vee_profile to numeric_parameter (as second length)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: vee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.14.3 vee_profile to numeric_parameter (as profile_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: vee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'profile angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.3.14.4 vee_profile to numeric_parameter (as profile_radius)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: vee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'profile radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.3.14.5 vee_profile to numeric_parameter (as tilt_angle)

AIM element: PATH
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: vee_profile <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tilt angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.4 library_reference UoF

5.1.4.1 BSU

#1: BSU is a Class_BSU

AIM element: externally_defined_class
Source: ISO 10303-224

#2: BSU is a Property_BSU

AIM element: externally_defined_general_property
Source: ISO 10303-224

#3: BSU is a Supplier_BSU

AIM element: organization
Source: ISO 10303-41

5.1.4.1.1 code

#1: BSU is a Class_BSU

AIM element: externally_defined_item.item_id
Source: ISO 10303-224
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 10303-224
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

5.1.4.2 Class_bsu

AIM element: externally_defined_class
Source: ISO 10303-224
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'}]

5.1.4.2.1 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 <-
library_class_version_assignment.items[i]
library_class_version_assignment <=
applied_external_identification_assignment <=
external_identification_assignment <=
{identification_assignment.role ->
identification_role
[identification_role.name = 'class version']}
identification_assignment

5.1.4.2.2 class_BSU to supplier_BSU (as defined_by)

AIM element: PATH
Source: ISO 10303-41
Reference Path: externally_defined_class <=
externally_defined_item
externally_defined_item.source ->
external_source =>

known_source
organization_item = known_source
organization_item <-
feature_based_pp_organization_assignment.items[i]
feature_based_pp_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
[organization_role.name = 'library supplier']}
organization_assignment
organization_assignment.assigned_organization ->
organization

5.1.4.3 Externally_defined_representation

AIM element: externally_defined_representation_with _parameters
Source: ISO 10303-224
Reference Path: externally_defined_representation_with_parameters<=
representation

5.1.4.3.1 externally_defined_representation to orientation (as placement_-location)

AIM element: placement
Source: ISO 10303-42
Reference Path: externally_defined_representation_with _parameters<=
representation
representation.items[i]->
representation_item=>
geometric_representation_item=>
placement

5.1.4.3.2 externally_defined_representation to library_part_assignment (as identified_by)

AIM element: PATH
Source: ISO 10303-224
Reference Path: externally_defined_representation_with_parameters<=
representation
classification_item=representation<-
applied_classification_assignment.items[i]
applied_classification_assignment

5.1.4.4 library_part_assignment

AIM element: applied_classification_assignment
Source: ISO 10303-224
Reference Path: applied_classification_assignment <=
classification_assignment

5.1.4.4.1 library_part_assignment to class_BSU (as definitional_class_BSU)

AIM element: PATH
Source: ISO 10303-224
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

5.1.4.4.2 library_part_assignment to property_value (as definitional_-property_value_pairs)

AIM element: PATH
Source: ISO 10303-43
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
externa_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_representation.definition
property_definition_representation
property_definition_representation.used_representation->
representation
{representation.name = 'property value'}

5.1.4.5 Property_bsu

AIM element: externally_defined_general_property
Source: ISO 10303-224
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'}]

5.1.4.5.1 version

AIM element: identification_asignment.assigned_id
Source: ISO 10303-41
Reference Path: externally_defined_general_property <=
externally_defined_item
external_identification_item=externally_defined_item
external_identification_item <-
library_property_version_assignment.items[i]
library_property_version_assignment <=
external_identification_assignment <=
identification_assignment
identification_asignment.assigned_id='property version'

5.1.4.5.2 property_BSU to class_BSU (as name_scope)

AIM element: PATH
Source: ISO 10303-224
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

5.1.4.6 Property_value

AIM element: representation
Source: ISO 10303-43
Reference Path: representation
{representation.name = ‘property value’}

5.1.4.6.1 property_value to property_BSU (as property_BSU)

AIM element: PATH
Source: ISO 10303-224
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

5.1.4.6.2 value_amount

#1: boolean

AIM element: descriptive_representation_item
Source: ISO 10303-41
Reference Path: representation
representation.items[i] ->
representation_item =>
descriptive_representation_item
{(descriptive_representation_item.description = ‘TRUE‘)
(descriptive_representation_item.description = ‘FALSE‘)}

#2: integer
#4: number
#5: real

AIM element: value_representation_item
Source: ISO 10303-41
Reference Path: representation
value_representation_item
value_representation_item.value_component
{value_representation_item.value_component = numeric_measure}

#3: logical

AIM element: descriptive_representation_item
Source: ISO 10303-41
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‘)}

#6: string

AIM element: descriptive_representation_item
Source: ISO 10303-41
Reference Path: representation
representation.items[i] ->
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

5.1.4.7 Supplier_bsu

AIM element: organization
Source: ISO 10303-41
Reference Path: externally_defined_class <=
externally_defined_item
externally_defined_item.source ->
external_source =>

known_source
organization_item = known_source
organization_item <-
feature_based_pp_organization_assignment.items[i]
feature_based_pp_organization_assignment <=
organization_assignment
{organization_assignment.role ->
organization_role
[organization_role.name = 'library supplier']}
organization_assignment
organization_assignment.assigned_organization ->
organization

5.1.5 manufacturing_feature UoF

5.1.5.1 Bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.1.1 Bevel_gear to numeric_parameter (as root_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'root angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.1.2 Bevel_gear to numeric_parameter (as tip_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tip angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.2 Boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: boss <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
(characterized_object.description = 'circular')
(characterized_object.description = 'complex')
(characterized_object.description = 'rectangular')}

5.1.5.2.1 boss to boss_top_condition (as top_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'top condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'boss top usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
boss_top

5.1.5.2.2 boss to linear_path (as boss_height)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'boss height occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'boss height']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'linear'}
shape_aspect =>
path_feature_component

5.1.5.2.3 boss to numeric_parameter (as fillet_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fillet radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.3 Catalogue_gear

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: externally_defined_feature_definition <=
[externally_defined_item
{externally_definied_item.item_id->
source_item
source_item=' external gear'}
{externally_definied_item.source->
external_source
external_source.source_id->
source_item
source_item=' external feature specification'}]
[{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'gear'}]

5.1.5.3.1 catalogue_gear to specification (as documentation)

#1: (if specification has zero constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_reference.items[i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document)
{document<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

#2: (if specification has one or more constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_usage_constraint_assignment.items[i]
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<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

5.1.5.4 Catalogue_knurl

AIM element: externally_defined_feature_definition
Source: ISO 10303-41
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:

```
externally_defined_feature_definition <=
[externally_defined_item
{externally_definied_item.item_id->
source_item
source_item=' external knurl'}
{externally_definied_item.source->
external_source
external_source.source_id->
source_item
source_item=' external feature specification'}]
[{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'knurl'}]
```

5.1.5.4.1 catalogue_knurl to specification (as documentation)

#1: (if specification has zero constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_reference.items[i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document)
{document<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

#2: (if specification has one or more constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_usage_constraint_assignment.items[i]
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<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

5.1.5.5 Catalogue_marking

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: externally_defined_feature_definition <=
[externally_defined_item
{externally_definied_item.item_id->
source_item
source_item=' external marking'}
{externally_definied_item.source->
external_source
external_source.source_id->
source_item
source_item=' external feature specification'}]
[{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'marking'}]

5.1.5.5.1 catalogue_marking to specification (as documentation)

#1: (if specification has zero constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_reference.items[i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document)
{document<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

#2: (if specification has one or more constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_usage_constraint_assignment.items[i]
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<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

5.1.5.6 Catalogue_thread

AIM element: externally_defined_feature_definition
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: externally_defined_feature_definition <=
[externally_defined_item
{externally_definied_item.item_id->
source_item
source_item=' external thread'}
{externally_definied_item.source->
external_source
external_source.source_id->
source_item
source_item=' external feature specification'}]
[{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'thread'}]

5.1.5.6.1 Catalogue_thread to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: externally_defined_feature_definition <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'major diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.6.2 catalogue_thread to specification (as documentation)

#1: (if specification has zero constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_reference.items[i]
applied_document_reference <=
document_reference
document_reference.assigned_document ->
document)
{document<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

#2: (if specification has one or more constrants)

AIM element: PATH
Reference Path: (externally_defined_feature_definition
document_reference_item = externally_defined_feature_definition
document_reference_item <-
applied_document_usage_constraint_assignment.items[i]
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<-
document_representation_type.represented_document
document_representation_type}
{document=>
document_file<=
characterized_object}

5.1.5.7 Chamfer

AIM element: chamfer
Source: ISO 10303-522
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: chamfer <=
transition_feature <=
shape_aspect

5.1.5.7.1 chamfer to face_shape_element (as chamfer_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: chamfer <=
transition_feature <=
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=shape_definition_representation}
property_definition_representation.used_representation->
representation
{representation.name= 'chamfer face'}
representation=>
shape_representation=>
face_shape_representation

5.1.5.7.2 chamfer to first_offset (as first_face_offset)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer <=
transition_feature <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
{shape_aspect
shape_aspect.description = 'first offset'}
shape_aspect =>
chamfer_offset

5.1.5.7.3 chamfer to second_chamfer_offset (as second_face_offset)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: chamfer <=
transition_feature <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
{shape_aspect
shape_aspect.description = 'second offset'}
shape_aspect =>
chamfer_offset

5.1.5.8 Circular_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: boss <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'circular'}

5.1.5.8.1 circular_boss to angle_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
taper

5.1.5.8.2 circular_boss to diameter_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
boss <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'diameter taper'}
shape_aspect =>
 taper
```


5.1.5.8.3 circular_boss to directed_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.8.4 circular_boss to circular_closed_profile (as circular_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'circular profile occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
circular_closed_profile

5.1.5.9 Circular_closed_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outside_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'closed circular boundary occurrence'}

5.1.5.9.1 circular_closed_shape_profile to complete_circular_profile (as closed_boundary)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'closed circular boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
shape_defining_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage'}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
circular_closed_profile

5.1.5.10 Circular_cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'circular cutout'}

5.1.5.10.1 circular_cutout to circular_closed_profile (as circular_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'enclosed boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
circular_closed_profile

5.1.5.11 Circular_offset_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: [shape_aspect <-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship =>
feature_component_relationship =>
pattern_offset_membership]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect=>
modified_pattern]

5.1.5.11.1 circular_offset_pattern to numeric_parameter (as angular_offset)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.11.2 circular_offset_pattern to numeric_parameter (as index_number)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'offset index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.12 Circular_omit_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: [shape_aspect <-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship =>
feature_component_relationship =>
pattern_omit_membership]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect=>
modified_pattern]

5.1.5.12.1 circular_omit_pattern to numeric_parameter (as omit_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'omit index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.13 Circular_pattern

AIM element: circular_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: circular_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.5.13.1 circular_pattern to circular_offset_pattern (as relocated_base_-feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: circular_pattern <=
replicate_feature <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
[shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship =>
pattern_offset_membership}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='base pattern'}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect<-
{shape_aspect=>
modified_pattern}
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='modified pattern'}
shape_aspect_relationship.related_shape_aspect->
shape_aspect]

5.1.5.13.2 circular_pattern to circular_omit_pattern (as missing_base_-feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: circular_pattern <=
replicate_feature <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
[shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship =>
pattern_omit_membership}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='base pattern'}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect<-
{shape_aspect=>
modified_pattern}
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='modified pattern'}
shape_aspect_relationship.related_shape_aspect->
shape_aspect]

5.1.5.13.3 circular_pattern to numeric_parameter (as angular_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:

```
circular_pattern <=
 replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'angular spacing'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}
```

5.1.5.13.4 circular_pattern to numeric_parameter (as base_feature_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: circular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.13.5 circular_pattern to numeric_parameter (as base_feature_rotation)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: circular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'base feature rotation'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.13.6 circular_pattern to numeric_parameter (as number_of_features)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: circular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of features'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.14 Closed_slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: slot <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.14.1 closed_slot to general_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'course of travel']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
{shape_aspect
shape_aspect.description = 'complex'}
path_feature_component

5.1.5.14.2 closed_slot to complete_circular_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
slot <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'course of travel']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
{shape_aspect
shape_aspect.description = 'complete circular'}
 path_feature_component
```

5.1.5.14.3 closed_slot to loop_slot_end_type (as end_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'end condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'slot end usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
{shape_aspect
shape_aspect.description = 'loop'}
slot_end

5.1.5.15 Compound_feature

AIM element: compound_feature
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: compound_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
shape_aspect.name = 'compound feature in solid'

5.1.5.15.1 feature_description

AIM element: characterized_object.description
Source: ISO 10303-41
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_object.description

5.1.5.15.2 feature_name

AIM element: characterized_object.name
Source: ISO 10303-41
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_object.name

5.1.5.15.3 compound_feature to compound_feature_element (as element)

#1: as element except for thread

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
{shape_aspect=>
composite_shape_aspect}
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
{shape_aspect_relationship=>
feature_component_relationship}
shape_aspect_relationship.related_shape_aspect->
shape_aspect=>
((instanced_feature)
(transition_feature))

#2: as element for thread only

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
{shape_aspect=>
composite_shape_aspect}
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
{shape_aspect_relationship=>
feature_component_relationship}
shape_aspect_relationship.related_shape_aspect->
shape_aspect=>
(applied_area)

5.1.5.16 Compound_feature_element

#1: as element except for thread

AIM element: (instanced_feature)
(transition_feature)
#2(applied_area)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: (instanced_feature <=
[feature_definition <=
characterized_object]
[shape_aspect])
(transition_feature <=
shape_aspect)

#2: as element for thread only

AIM element: (applied_area)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: (applied_area <=
shape_aspect)

5.1.5.16.1 compound_feature_element to machining_feature (as element)

AIM element: IDENTICAL MAPPING

5.1.5.16.2 compound_feature_element to transition_feature (as element)

AIM element: IDENTICAL MAPPING

5.1.5.17 Compound_feature_relationship

AIM element: shape_aspect_relationship
Source: ISO 10303-41
Reference Path: {shape_aspect_relationship
shape_aspect_relationship.name = 'compound feature ordering'}

5.1.5.17.1 compound_feature_relationship to compound_feature_element (as predecessor)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
(instanced_feature)
(transition_feature)

5.1.5.17.2 compound_feature_relationship to compound_feature_element (as successor)

AIM element: PATH
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(instanced_feature)
(transition_feature)

5.1.5.18 Constant_radius_edge_round

AIM element: edge_round
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
{shape_aspect
shape_aspect.description = 'constant radius'}

5.1.5.18.1 constant_radius_edge_round to numeric_parameter (as first_face_-offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.18.2 constant_radius_edge_round to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.18.3 constant_radius_edge_round to numeric_parameter (as second_-face_offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: edge_round <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.19 Constant_radius_fillet

AIM element: fillet
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: fillet <=
transition_feature <=
shape_aspect
{shape_aspect
shape_aspect.description = 'constant radius'}

5.1.5.19.1 constant_radius_fillet to numeric_parameter (as first_face_offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'first offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.19.2 constant_radius_fillet to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.19.3 constant_radius_fillet to numeric_parameter (as second_face_-offset)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'second offset'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.20 Counterbore_hole

AIM element: composite_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object
characterized_object.description = 'counterbore']
instanced_feature <=
shape_aspect

5.1.5.20.1 counterbore_hole to round_hole (as larger_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: composite_hole <=
compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
{shape_aspect=>
composite_shape_aspect}
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.name = 'large hole']}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature <=
feature_definition =>
round_hole

5.1.5.20.2 counterbore_hole to round_hole (as smaller_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: composite_hole <=
compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
{shape_aspect=>
composite_shape_aspect}
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.name = 'small hole']}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature <=
feature_definition =>
round_hole

5.1.5.21 Countersunk_hole

AIM element: composite_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object
characterized_object.description = 'countersunk']
instanced_feature <=
shape_aspect

5.1.5.21.1 countersunk_hole to round_hole (as constant_diameter_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: composite_hole <=
compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
{shape_aspect=>
composite_shape_aspect}
shape_aspect<-
shape_aspect_relationship.relating_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.name = 'constant diameter hole']}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature <=
feature_definition =>
round_hole

5.1.5.21.2 countersunk_hole to round_hole (as tapered_hole)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: composite_hole <=
compound_feature <=
feature_definition <=
characterized_object
characterized_definition=characterized_object<-
property_definition.definition
property_definition
property_definition=>
product_definition_shape<-
shape_aspect.of_shape
shape_aspect
shape_aspect=>
composite_shape_aspect
composite_shape_aspect.component_relationships->
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature <=
feature_definition =>
round_hole

{round_hole <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
taper}

5.1.5.22 Cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
(characterized_object.description = 'circular cutout')
(characterized_object.description = 'complex cutout')}

5.1.5.22.1 cutout to through_pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'pocket bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'through')}
shape_aspect =>
pocket_bottom

5.1.5.23 Defined_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.24 Defined_marking

AIM element: marking
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: marking <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.24.1 defined_marking to descriptive_parameter (as font_name)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'font name'}
representation_item =>
descriptive_representation_item

5.1.5.24.2 defined_marking to descriptive_parameter (as special_instructions)

AIM element: PATH

Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)

Reference Path:

```
marking <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'special instructions'}
representation_item =>
descriptive_representation_item
```


5.1.5.24.3 defined_marking to numeric_parameter (as character_height)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'character height'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.24.4 defined_marking to numeric_parameter (as character_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: marking <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'character spacing'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.25 Defined_thread

AIM element: thread
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: thread <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.25.1 defined_thread to numeric_parameter (as crest)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'crest'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.25.2 defined_thread to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'major diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.25.3 defined_thread to numeric_parameter (as minor_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'minor diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.25.4 defined_thread to numeric_parameter (as pitch_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.26 Diagonal_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: turned_knurl <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'diagonal'}

5.1.5.26.1 diagonal_knurl to descriptive_parameter (as helix_hand)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'helix hand'}
representation_item =>
descriptive_representation_item

5.1.5.26.2 diagonal_knurl to numeric_parameter (as helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'helix angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.27 Diamond_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: turned_knurl <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'diamond'}

5.1.5.27.1 diamond_knurl to numeric_parameter (as helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'helix angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.28 Edge_round

AIM element: edge_round
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: edge_round <=
transition_feature <=
shape_aspect

5.1.5.28.1 edge_round to face_shape_element (as edge_round_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path:
```
edge_round <=
 transition_feature <=
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=shape_definition_representation}
property_definition_representation.used_representation->
representation
{representation.name= 'edge round face'}
representation=>
shape_representation=>
face_shape_representation
```

5.1.5.28.2 edge_round to face_shape_element (as first_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path:
```
edge_round <=
 transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'first face shape'}
representation =>
shape_representation=>
face_shape_representation
```

5.1.5.28.3 edge_round to face_shape_element (as second_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: edge_round <=
 transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'second face shape'}
representation =>
shape_representation=>
 face_shape_representation

5.1.5.29 Fillet

AIM element: fillet
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: fillet <=
 transition_feature <=
shape_aspect

5.1.5.29.1 fillet to face_shape_element (as fillet_face)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: fillet <=
transition_feature <=
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=shape_definition_representation}
property_definition_representation.used_representation->
representation
{representation.name= 'fillet face'}
representation=>
shape_representation=>
face_shape_representation

5.1.5.29.2 fillet to face_shape_element (as first_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'first face shape'}
representation =>
shape_representation=>
face_shape_representation

5.1.5.29.3 fillet to face_shape_element (as second_face_shape)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: fillet <=
transition_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'second face shape'}
representation =>
shape_representation=>
face_shape_representation

5.1.5.30 Gear

AIM element: (gear)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object)
(externally_defined_feature_definition <=
[externally_defined_item]
[{feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
(characterized_object.description = 'helical bevel gear')
(characterized_object.description = 'straight bevel gear')
(characterized_object.description = 'spur gear')
(characterized_object.description = 'helix gear')}])

5.1.5.30.1 internal_or_external_gear

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:
```
( gear <=)
( externally_defined_feature_definition <=)
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch type'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'internal')
(descriptive_representation_item.description = 'external')}
```

5.1.5.30.2 module_or_diametral_pitch

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch type'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'module')
(descriptive_representation_item.description = 'diametral pitch')}

5.1.5.30.3 gear to numeric_parameter (as normal_attribute)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'normal_attribute'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.30.4 gear to numeric_parameter (as nominal_tooth_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'nominal tooth depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.30.5 gear to numeric_parameter (as reference_pressure_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'reference pressure angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.30.6 gear to numeric_parameter (as profile_shift)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'profile shift'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.30.7 gear to numeric_parameter (as number_of_teeth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of teeth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
ratio_measure_with_unit}

5.1.5.30.8 gear to numeric_parameter (as face_width)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:

```
( gear <=)
( externally_defined_feature_definition <=)
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'face width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}
```


5.1.5.30.9 gear to numeric_parameter (as root_fillet_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'root fillet radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.30.10 gear to numeric_parameter (as tip_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tip diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.30.11 gear to shape (as applied_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition =>
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape

5.1.5.31 General_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: boss <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'complex'}

5.1.5.31.1 general_boss to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
taper

5.1.5.31.2 general_boss to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.31.3 general_boss to closed_profile (as enclosed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:
boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'enclosed boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(closed_path_profile)
(ngon_closed_profile)

5.1.5.32 General_cutout

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:
pocket <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'complex cutout'}

5.1.5.32.1 general_cutout to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(linear_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.33 General_outside_profile

AIM element:	outside_profile
Source:	ISO 10303-522
Rules:	machining_feature_life_cycle – (See 5.2.4.17) shape_aspect_subtype_exclusiveness – (See 5.2.4.34) subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:	outside_profile <= { feature_definition => instanced_feature} feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'outside boundary'}

5.1.5.33.1 general_outside_profile to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'outside boundary'}
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(linear_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.34 General_pattern

AIM element: feature_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: feature_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.5.34.1 general_pattern to orientation (as feature_placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: feature_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
characterized_definition=charcterized_object
characterized_definition<-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'base feature placement'}
representation_item =>
geometric_representation_item =>
placement

5.1.5.35 General_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'complex'}

5.1.5.35.1 general_pocket to boss (as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.35.2 general_pocket to protrusion(as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
protrusion

5.1.5.35.3 general_pocket to profile (as boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'boundary'}
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.36 General_removal_volume

AIM element: removal_volume
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: removal_volume <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.36.1 general_removal_volume to shape_element (as removal_volume)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: removal_volume <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <=
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'shape volume occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'volume shape usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'volume shape'}
shape_aspect

5.1.5.37 General_revolution

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: revolved_profile <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'open profile'}

5.1.5.37.1 general_revolution to general_open_profile (as outer_edge_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: revolved_profile
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'outer edge shape occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'outer edge shape'}
shape_aspect =>
open_path_profile

5.1.5.38 General_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outside_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'complex boundary occurrence'}

5.1.5.38.1 general_shape_profile to path (as profile_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'complex boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
shape_defining_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage'}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'boundary'}
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.39 Groove

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: revolved_profile <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'groove'}

5.1.5.39.1 groove to open_profile (as sweep)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: revolved_profile <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'sweep occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'sweep'}
shape_aspect =>
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.40 Helical_bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'helix bevel gear'}

5.1.5.40.1 left_or_right_hand_tooth

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch type'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'left hand tooth')
(descriptive_representation_item.description = 'right hand tooth')}

5.1.5.40.2 helical_gear to numeric_parameter (as reference_helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'reference helix angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.41 Helical_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'helix gear'}

5.1.5.41.1 left_or_right_hand_tooth

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch type'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'left hand tooth')
(descriptive_representation_item.description = 'right hand tooth')}

5.1.5.41.2 helical_gear to numeric_parameter (as reference_helix_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (gear <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'reference helix angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.42 Hole

#1: if hole is of type round_hole

AIM element: (round_hole)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: round_hole <=
{feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

#1: if hole is of type countersunk_hole or counterbore_hole

AIM element: (composite_hole)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: composite_hole <=
compound_feature <=
feature_definition=>
[feature_definition <=
characterized_object]
instanced_feature <=
[shape_aspect]

5.1.5.43 Knurl

AIM element: (turned_knurl)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (turned_knurl <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object)
(externally_defined_feature_definition <=
[externally_defined_item]
[{feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'knurl'}])

5.1.5.43.1 knurl to partial_area_definition (as partial_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (turned_knurl <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape
shape_aspect.of_shape->
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'applied area usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
applied_area

5.1.5.43.2 knurl to shape_element (as applied_shape)

AIM element: PATH
Source: ISO 10303-522
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (turned_knurl <=)
(externally_defined_feature_definition <=)
feature_definition =>
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape
shape_aspect.of_shape->
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'applied shape']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect

5.1.5.44 Machining_feature

AIM element: instanced_feature
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: instanced_feature <=
[shape_aspect]
[feature_definition <=
characterized_object]

5.1.5.44.1 usage_name

AIM element: shape_aspect.description
Source: ISO 10303-41
Reference Path: instanced_feature <=
shape_aspect
shape_aspect.description

5.1.5.44.2 machining_feature to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: instanced_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
{property_definition=>
product_definition_shape}
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.5.45 Manufacturing_feature

AIM element: characterized_object
Source: ISO 10303-41

5.1.5.46 Manufacturing_feature_group

AIM element: group
Source: ISO 10303-41

5.1.5.46.1 group_description

AIM element: group.description
Source: ISO 10303-41

5.1.5.46.2 group_name

AIM element: group.name
Source: ISO 10303-41

5.1.5.46.3 manufacturing_feature_group to manufacturing_feature(as feature_groups)

AIM element: PATH
Reference Path: group<-
group_assignment.assigned_group
group_assignment=>
applied_group_assignment
applied_group_assignment.items->
group_item
(group_item=instanced_feature)
(group_item=replicate_feature)
(group_item=transition_feature)

5.1.5.46.4 manufacturing_feature_group to manufacturing_feature_group(as feature_groups)

AIM element: PATH
Reference Path: group<-
group_relationship.related_group
group_relationship
group_relationship.relating_group->
group

5.1.5.47 Marking

AIM element: (marking)
(externally_defined_feature_definition)
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (marking <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object)
(externally_defined_feature_definition <=
[externally_defined_item]
[{feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'marking'}])

5.1.5.47.1 marking to shape (as applied_to_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (marking <=)
(externally_defined_feature_definition <=)
feature_definition =>
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape
shape_aspect.of_shape->
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'applied shape']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect

5.1.5.47.2 marking to descriptive_parameter (as text)

AIM element: PATH
Reference Path: (marking <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'marking text'}
representation_item =>
descriptive_representation_item

5.1.5.48 Multi_axis_feature

AIM element: instanced_feature
Source: ISO 10303-522
Reference Path: (boss<=)
(outside_profile<=)
(removal_volume<=)
(round_hole<=)
(flat_face<=)
(pocket<=)
(protrusion<=)
(rib_top<=)
(rounded_end<=)
(slot<=)
(step<=)
instanced_feature

5.1.5.48.1 maximum_feature_limit

AIM element: PATH
Reference Path: instanced_feature <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
{property_definition=>
product_definition_shape}
property_definition <-
property_definition_representation.definition
property_definition_representation
{property_definition_representation =>
shape_definition_representation}
property_definition_representation.used_representation->
representation=>
{representation.name='maximum feature limit'}
shape_representation=>
planar_shape_representation

5.1.5.49 Open_slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: slot <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object

5.1.5.49.1 open_slot to path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'course of travel']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 path_feature_component

5.1.5.49.2 open_slot to slot_end_type (as end_conditions)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'end condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'slot end usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
slot_end

5.1.5.50 Outer_diameter

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outer_round <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'outer diameter'}

5.1.5.50.1 outer_diameter to angle_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'reduced size occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'reduced size']
[shape_aspect_relationship.description = 'taper usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
taper

5.1.5.50.2 outer_diameter to diameter_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
outer_round <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'reduced size occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'reduced size']
[shape_aspect_relationship.description = 'taper usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'diameter taper'}
shape_aspect =>
 taper
```


5.1.5.50.3 outer_diameter to directed_taper (as reduced_size)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'reduced size occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'reduced size']
[shape_aspect_relationship.description = 'taper usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.50.4 outer_diameter to numeric_parameter (as diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.50.5 outer_diameter to numeric_parameter (as feature_length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.51 Outer_diameter_to_shoulder

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outer_round <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'outer diameter to shoulder'}

5.1.5.51.1 outer_diameter_to_shoulder to numeric_parameter (as diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.51.2 outer_diameter_to_shoulder to numeric_parameter (as feature_-length)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'feature length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.51.3 outer_diameter_to_shoulder to vee_profile (as v_shape_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outer_round <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'v-shape boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'v-shape'}
shape_aspect =>
vee_profile

5.1.5.52 Outer_round

AIM element: outer_round
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outer_round <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.53 Partial_circular_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outside_profile <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'partial circular boundary occurrence'}

5.1.5.53.1 partial_circular_shape_profile to partial_circular_profile (as open_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'partial circular boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
 shape_defining_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage'}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
partial_circular_profile

5.1.5.54 Planar_face

AIM element: flat_face
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: flat_face <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.54.1 planar_face to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.54.2 planar_face to linear_path (as course_of_travel)

AIM element: PATH

Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)

Reference Path:

```
flat_face <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'course of travel']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'linear'}
shape_aspect =>
 path_feature_component
```

5.1.5.54.3 planar_face to linear_profile (as removal_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:
flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'removal boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'removal boundary']
[shape_aspect_relationship.description = 'profile usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
linear_profile

5.1.5.54.4 planar_face to linear_profile (as face_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'enclosed boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'boundary']
[shape_aspect_relationship.description = 'profile usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(circular_closed_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(closed_path_profile)

5.1.5.54.5 planar_face to numeric_parameter (as removal_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: flat_face <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'removal depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.54.6 planar_face to boss (as volume_not_removed)

AIM element: PATH
Reference Path: flat_face <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.54.7 planar_face to protrusion(as volume_not_removed)

AIM element:	PATH
Reference Path:	flat_face <= feature_ definition <= characterized_ object characterized_ definition= characterized_ object characterized_ definition <- property_ definition. definition property_ definition => product_ definition_ shape< - shape_ aspect. of_ shape {shape_ aspect=> composite_ shape_ aspect} shape_ aspect< - shape_ aspect_ relationship. relating_ shape_ aspect {shape_ aspect_ relationship. description= 'uncut volume'} {shape_ aspect_ relationship=> feature_ component_ relationship} shape_ aspect_ relationship shape_ aspect_ relationship. related_ shape_ aspect-> shape_ aspect=> instanced_ feature=> feature_ definition=> protrusion

5.1.5.55 Pocket

AIM element:	pocket
Source:	ISO 10303-522
Rules:	machining_feature_life_cycle – (See 5.2.4.17) shape_aspect_subtype_exclusiveness – (See 5.2.4.34) subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:	pocket <= { feature_definition => instanced_feature} feature_definition <= characterized_object

5.1.5.55.1 pocket to numeric_parameter (as base_radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:
```
pocket <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fillet radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}
```


5.1.5.55.2 pocket to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
taper

5.1.5.55.3 pocket to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.55.4 pocket to linear_path (as pocket_depth)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'pocket depth occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'pocket depth']
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'linear'}
shape_aspect =>
path_feature_component

5.1.5.55.5 pocket to pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'pocket bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'planar')
(shape_aspect.description = 'complex')}
shape_aspect =>
 pocket_bottom

5.1.5.55.6 pocket to through_pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'pocket bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'through')}
shape_aspect =>
pocket_bottom

5.1.5.56 Profile_feature

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: (outside_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object)

5.1.5.56.1 Profile_feature to linear_profile (as profile_swept_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:
```
outside_profile<=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
(shape_aspect.description = 'outside boundary'),
(shape_aspect.description = 'complex boundary occurrence'),
(shape_aspect.description = 'partial circular boundary occurrence'),
(shape_aspect.description = 'closed circular boundary occurrence'),
(shape_aspect.description = 'open rectangular boundary occurrence'),
(shape_aspect.description = 'closed rectangular boundary occurrence')}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'profile swept shape']
[shape_aspect_relationship.description = 'path feature component usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect
shape_aspect.description='linear'
shape_aspect =>
 path_feature_component
```

5.1.5.57 Protrusion

AIM element: protrusion
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:
```
protrusion <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
```

5.1.5.57.1 protrusion to shape_element (as shape_volume)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: protrusion <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'shape volume occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'volume shape usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'volume shape'}
shape_aspect

5.1.5.58 Recess

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'recess'}

5.1.5.58.1 recess to pocket_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'pocket bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'planar')
(shape_aspect.description = 'complex')}
shape_aspect =>
pocket_bottom

5.1.5.58.2 recess to closed_profile (as fillet boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(circular_closed_profile)
(closed_path_profile)
(ngon_closed_profile)
(rectangular_closed_profile)
(linear_profile)
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.58.3 recess to boss (as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.58.4 recess to protrusion(as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
protrusion

5.1.5.59 Rectangular_boss

AIM element: boss
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: boss <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'rectangular'}

5.1.5.59.1 rectangular_boss to angle_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
boss <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
 taper
```

5.1.5.59.2 rectangular_boss to directed_taper (as change_in_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.59.3 rectangular_boss to closed_profile (as enclosed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: boss <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'rectangular profile occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 rectangular_closed_profile

5.1.5.60 Rectangular_closed_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'closed rectangular'}

5.1.5.60.1 rectangular_closed_pocket to boss (as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.60.2 rectangular_closed_pocketto protrusion (as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
protrusion

5.1.5.60.3 rectangular_closed_pocket to rectangular_closed_profile (as closed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: pocket <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'closed boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 rectangular_closed_profile

5.1.5.61 Rectangular_closed_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outside_profile <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'closed rectangular boundary occurrence'}

5.1.5.61.1 rectangular_closed_shape_profile to rectangular_closed_profile (as closed_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'closed rectangular boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
shape_defining_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage'}
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
rectangular_closed_profile

5.1.5.62 Rectangular_offset_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: 5.2.4.33
Reference Path: [shape_aspect <-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship =>
feature_component_relationship =>
pattern_offset_membership]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect=>
modified_pattern]

5.1.5.62.1 rectangular_offset_pattern to direction_element (as offset_-direction)

AIM element: PATH
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: shape_aspect <-
shape_definition = shape_aspect
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'offset direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.62.2 rectangular_offset_pattern to numeric_parameter (as column_-index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'column index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.62.3 rectangular_offset_pattern to numeric_parameter (as offset_-distance)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path:

```
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
 shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'offset distance'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}
```

5.1.5.62.4 rectangular_offset_pattern to numeric_parameter (as row_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'row index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure}

5.1.5.63 Rectangular_omit_pattern

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: [shape_aspect <-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship =>
feature_component_relationship =>
pattern_omit_membership]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect=>
modified_pattern]

5.1.5.63.1 rectangular_omit_pattern to numeric_parameter (as column_-index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'column index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.63.2 rectangular_omit_pattern to numeric_parameter (as row_index)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'row index'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.64 Rectangular_open_pocket

AIM element: pocket
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: pocket <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'open rectangular'}

5.1.5.64.1 rectangular_open_pocket to boss (as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.64.2 rectangular_open_pocketto protrusion(as volume_not_removed)

AIM element: PATH
Reference Path: pocket <=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
protrusion

5.1.5.64.3 rectangular_open_pocket to square_u_profile (as open_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:
```
pocket <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'open boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 square_u_profile
```

5.1.5.65 Rectangular_open_shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:
```
outside_profile <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'open rectangular boundary occurrence'}
```

5.1.5.65.1 rectangular_open_shape_profile to square_U_profile (as open_-boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'open rectangular boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
shape_defining_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage'}
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
square_u_profile

5.1.5.66 Rectangular_pattern

AIM element: rectangular_pattern
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rectangular_pattern <=
replicate_feature <=
{feature_definition=>
instanced_feature}
feature_definition<=
characterized_object

5.1.5.66.1 rectangular_pattern to direction_element (as column_layout_-direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'column layout direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.66.2 rectangular_pattern to direction_element (as row_layout_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'row layout direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.66.3 rectangular_pattern to numeric_parameter (as column_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'column spacing'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.66.4 rectangular_pattern to numeric_parameter (as columns)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of columns'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.66.5 rectangular_pattern to numeric_parameter (as row_spacing)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'row spacing'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.66.6 rectangular_pattern to numeric_parameter (as rows)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition<=
characterized_object
shape_definition = characterized_object
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of rows'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.66.7 rectangular_pattern to rectangular_offset_pattern (as relocated_-base_feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: rectangular_pattern <=
 replicate_feature <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
[shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
 feature_component_relationship =>
 pattern_offset_membership}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
 feature_component_relationship}
{shape_aspect_relationship.description='base pattern'}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect<-
{shape_aspect=>
modified_pattern}
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
 feature_component_relationship}
{shape_aspect_relationship.description='modified pattern'}
shape_aspect_relationship.related_shape_aspect->
shape_aspect]

5.1.5.66.8 rectangular_pattern to rectangular_omit_pattern (as missing_-base_feature)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: rectangular_pattern <=
replicate_feature <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
[shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship =>
pattern_omit_membership}
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect]
[shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='base pattern'}
shape_aspect_relationship.relating_shape_aspect->
shape_aspect<-
{shape_aspect=>
modified_pattern}
shape_aspect_relationship.relating_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship}
{shape_aspect_relationship.description='modified pattern'}
shape_aspect_relationship.related_shape_aspect->
shape_aspect]

5.1.5.67 Replicate_base

AIM element: feature_component_relationship
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: feature_component_relationship <=
shape_aspect_relationship

5.1.5.67.1 replicate_base to machining_feature (as base_feature)

AIM element: PATH
Reference Path: feature_component_relationship <=
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature

5.1.5.67.2 replicate_base to replicate_feature (as base_feature)

AIM element: PATH
Reference Path: feature_component_relationship <=
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
instanced_feature<=
feature_definition<=
replicate_feature

5.1.5.68 Replicate_feature

AIM element: replicate_feature
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: replicate_feature <=
{feature_definition<=
instanced_feature}
feature_definition<=
characterized_object

5.1.5.68.1 replicate_feature to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: replicate_feature <=
feature_definition=>
instanced_feature<=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.5.68.2 replicate_feature to replicate_base (as replicate_base_feature)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: replicate_feature <=
feature_definition=>
instanced_feature<=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship =>
feature_component_relationship
{shape_aspect_relationship.name='pattern basis'}

5.1.5.69 Revolved_feature

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: revolved_profile <=
{ feature_definition = instanced_feature}
feature_definition <=
characterized_object

5.1.5.69.1 revolved_feature to direction_element (as material_side)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.69.2 revolved_feature to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.70 Revolved_flat

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: revolved_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'flat'}

5.1.5.70.1 revolved_flat to linear_profile (as flat_edge_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'flat edge shape occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'flat edge shape'}
shape_aspect =>
linear_profile

5.1.5.71 Revolved_round

AIM element: revolved_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: revolved_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'round'}

5.1.5.71.1 revolved_round to partial_circular_profile (as rounded_edge_-shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: revolved_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'rounded edge shape occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'rounded edge shape'}
shape_aspect =>
partial_circular_profile

5.1.5.72 Rip_top

AIM element: rib_top
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: rib_top <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.72.1 rib_top to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path:
ribtop <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name= 'removal direction'}
representation =>
shape_representation =>
direction_shape_representation

5.1.5.72.2 rib_top to rib_top_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: ribtop <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'rib top condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'ribtop usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'planar')
(shape_aspect.description = 'complex')}
shape_aspect =>
ribtop_floor

5.1.5.73 Round_hole

AIM element: round_hole
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: round_hole <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.73.1 round_hole to angle_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
round_hole <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'angle taper'}
shape_aspect =>
 taper
```

5.1.5.73.2 round_hole to blind_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: round_hole <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'hole bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'conical')
(shape_aspect.description = 'flat')
(shape_aspect.description = 'flat with radius')
(shape_aspect.description = 'flat with taper')
(shape_aspect.description = 'spherical')}
shape_aspect =>
 hole_bottom

5.1.5.73.3 round_hole to circular_closed_profile (as diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:
```
round_hole <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.name = 'diameter']
[shape_aspect_relationship.description = 'profile usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 circular_closed_profile
```


5.1.5.73.4 round_hole to diameter_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: round_hole <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'diameter taper'}
shape_aspect =>
taper

5.1.5.73.5 round_hole to directed_taper (as change_in_diameter)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: round_hole <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'change in diameter occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'taper usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'directed taper'}
shape_aspect =>
taper

5.1.5.73.6 round_hole to linear_path (as hole_depth)

AIM element:	PATH
Rules:	shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33) shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:	round_hole <= feature_definition <= characterized_object characterized_definition = characterized_object characterized_definition <- property_definition.definition property_definition => product_definition_shape <- shape_aspect.of_shape {shape_aspect shape_aspect.description = 'hole depth occurrence'} shape_aspect <- shape_aspect_relationship.related_shape_aspect {[shape_aspect_relationship => shape_defining_relationship] [shape_aspect_relationship [shape_aspect_relationship.name = 'hole depth'] [shape_aspect_relationship.description = 'path feature component usage']]} shape_aspect_relationship shape_aspect_relationship.relating_shape_aspect -> {shape_aspect shape_aspect.description = 'linear'} shape_aspect => path_feature_component

5.1.5.73.7 round_hole to through_bottom_condition (as bottom_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: round_hole <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'bottom condition occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 feature_component_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'hole bottom usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'through'}
shape_aspect =>
 hole_bottom

5.1.5.74 Rounded_end

AIM element: rounded_end
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: rounded_end <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object

5.1.5.74.1 rounded_end to linear_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
rounded_end <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'linear'}
shape_aspect =>
 path_feature_component
```

5.1.5.74.2 rounded_end to partial_circular_profile (as partial_circular_-boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: rounded_end <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'partial circular boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.description = 'profile usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
partial_circular_profile

5.1.5.75 Shape_profile

AIM element: outside_profile
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: outside_profile <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.75.1 shape_profile to profile_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
(shape_aspect.description = 'complex boundary occurrence')
(shape_aspect.description = 'partial circular boundary occurrence')
(shape_aspect.description = 'closed circular boundary occurrence')
(shape_aspect.description = 'open rectangular boundary occurrence')
(shape_aspect.description = 'closed rectangular boundary occurrence')}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile floor usage'}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'planar')
(shape_aspect.description = 'complex')}
shape_aspect =>
profile_floor

5.1.5.75.2 shape_profile to through_profile_floor (as floor_condition)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
(shape_aspect.description = 'complex boundary occurrence'),
(shape_aspect.description = 'partial circular boundary occurrence'),
(shape_aspect.description = 'closed circular boundary occurrence'),
(shape_aspect.description = 'open rectangular boundary occurrence'),
(shape_aspect.description = 'closed rectangular boundary occurrence')}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{shape_aspect_relationship =>
feature_component_relationship
shape_aspect_relationship
shape_aspect_relationship.description = 'profile floor usage'}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
(shape_aspect.description = 'through')}
shape_aspect =>
profile_floor

5.1.5.75.3 shape_profile to direction_element (as removal_direction)

AIM element: PATH
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: outside_profile <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
{property_definition=>
product_definition_shape}
{product_definition_shape <-
shape_aspect.of_shape
shape_aspect
(shape_aspect.description = 'complex boundary occurrence'),
(shape_aspect.description = 'partial circular boundary occurrence'),
(shape_aspect.description = 'closed circular boundary occurrence'),
(shape_aspect.description = 'open rectangular boundary occurrence'),
(shape_aspect.description = 'closed rectangular boundary occurrence')}
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
{representation
representation.name = 'removal direction'}
representation =>
shape_representation=>
direction_shape_representation

5.1.5.76 Slot

AIM element: slot
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: slot <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.76.1 slot to open_profile (as sweep_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: slot <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'swept shape occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
(open_path_profile)
(partial_circular_profile)
(rounded_u_profile)
(square_u_profile)
(tee_profile)
(vee_profile)

5.1.5.77 Spherical_cap

AIM element: spherical_cap
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: spherical_cap <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.77.1 spherical_cap to numeric_parameter (as internal_angle)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: spherical_cap <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'internal angle'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.77.2 spherical_cap to numeric_parameter (as radius)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: spherical_cap <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'radius'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.78 Spur_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'spur gear'}

5.1.5.79 Step

AIM element: step
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: step <=
{ feature_definition =>
 instanced_feature}
feature_definition <=
characterized_object

5.1.5.79.1 step to linear_path (as course_of_travel)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: step <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'course of travel occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
[shape_aspect_relationship.description = 'path feature component usage']]}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
{shape_aspect
shape_aspect.description = 'linear'}
shape_aspect =>
 path_feature_component

5.1.5.79.2 step to vee_profile (as removal_boundary)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path:

```
 step <=
 feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape <-
shape_aspect.of_shape
{shape_aspect
shape_aspect.description = 'removal boundary occurrence'}
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
 shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'profile usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
 vee_profile
```

5.1.5.79.3 step to boss (as volume_not_removed)

AIM element: PATH
Reference Path: step<=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
boss

5.1.5.79.4 step to protrusion(as volume_not_removed)

AIM element: PATH
Reference Path: step<=
feature_ definition <=
characterized_ object
characterized_ definition= characterized_ object
characterized_ definition <-
property_ definition. definition
property_ definition =>
product_ definition_ shape< -
shape_ aspect. of_ shape
{shape_ aspect=>
composite_ shape_ aspect}
shape_ aspect< -
shape_ aspect_ relationship. relating_ shape_ aspect
{shape_ aspect_ relationship. description= 'uncut volume'}
{shape_ aspect_ relationship=>
feature_ component_ relationship}
shape_ aspect_ relationship
shape_ aspect_ relationship. related_ shape_ aspect->
shape_ aspect=>
instanced_ feature=>
feature_ definition=>
protrusion

5.1.5.80 Straight_bevel_gear

AIM element: gear
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: gear <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object
{characterized_object
characterized_object.description = 'straight bevel gear'}

5.1.5.81 Straight_knurl

AIM element:	turned_knurl
Source:	ISO 10303-522
Rules:	machining_feature_life_cycle – (See 5.2.4.17) shape_aspect_subtype_exclusiveness – (See 5.2.4.34) subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path:	turned_knurl <= { feature_definition => instanced_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'straight'}

5.1.5.82 Thread

AIM element:	(thread) (externally_defined_feature_definition)
Source:	ISO 10303-522
Rules:	machining_feature_life_cycle – (See 5.2.4.17) shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path:	(thread <= { feature_definition => instanced_feature} feature_definition <= characterized_object) (externally_defined_feature_definition <= [externally_defined_item] [{feature_definition => instanced_feature} feature_definition <= characterized_object {characterized_object characterized_object.description = 'thread'}])

5.1.5.82.1 thread to descriptive_parameter (as fit_class)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fit class'}
representation_item =>
descriptive_representation_item

5.1.5.82.2 thread to descriptive_parameter (as fit_class_2)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'fit class 2'}
representation_item =>
descriptive_representation_item

5.1.5.82.3 thread to descriptive_parameter (as form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'form'}
representation_item =>
descriptive_representation_item

5.1.5.82.4 thread to descriptive_parameter (as qualifier)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'qualifier'}
representation_item =>
descriptive_representation_item

5.1.5.82.5 inner_or_outer_thread

AIM element: descriptive_representation_item.description
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'thread side'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description = 'internal')
(descriptive_representation_item.description = 'external')}

5.1.5.82.6 thread to numeric_parameter (as number_of_threads)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of threads'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
ratio_measure_with_unit}

5.1.5.82.7 thread to numeric_parameter (as nominal_size)

AIM element: PATH

Rules:
dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)

Reference Path:
(thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'nominal size'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.82.8 thread to partial_area_definition (as partial_profile)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape
shape_aspect.of_shape->
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'applied area usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
applied_area

5.1.5.82.9 thread to shape (as applied_shape)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition =>
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape

5.1.5.82.10 thread to descriptive_parameter (as thread_hand)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'hand'}
{representation_item
representation_item.description = 'left'
representation_item.description = 'right'}
representation_item =>
descriptive_representation_item

5.1.5.82.11 thread to thread_runout (as runout)

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: (thread <=)
(externally_defined_feature_definition <=)
feature_definition <=
characterized_object<-
property_definition.definition
property_definition=>
product_definition_shape
shape_aspect.of_shape->
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
{[shape_aspect_relationship =>
shape_defining_relationship]
[shape_aspect_relationship
shape_aspect_relationship.description = 'thread runout usage']}
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
thread_runout

5.1.5.83 Thread_runout

AIM element: thread_runout
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: thread_runout<=
shape_aspect
{shape_aspect
shape_aspect.of_shape->
product_definition_shape}

5.1.5.83.1 Thread_runout to numeric_parameter (as length_of_runout)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread_runout <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'length of runout'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.83.2 Thread_runout to descriptive_parameter (as included_or_extra)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread_runout <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'included_or_extra'}
representation_item =>
descriptive_representation_item
(descriptive_representation_item.description='included')
(descriptive_representation_item.description='extra')

5.1.5.83.3 Thread_runout to descriptive_parameter (as pitch_or_dimension)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: thread_runout <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'pitch or dimension'}
representation_item =>
descriptive_representation_item
(descriptive_representation_item.description='pitch')
(descriptive_representation_item.description='dimension')

5.1.5.84 Transition_feature

AIM element: transition_feature
Source: ISO 10303-522
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
transition_feature_life_cycle – (See 5.2.4.37)
transition_feature_on_part_boundary – (See 5.2.4.38)
Reference Path: transition_feature <=
shape_aspect

5.1.5.85 Turned_knurl

AIM element: turned_knurl
Source: ISO 10303-522
Rules: machining_feature_life_cycle – (See 5.2.4.17)
shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: turned_knurl <=
{ feature_definition =>
instanced_feature}
feature_definition <=
characterized_object

5.1.5.85.1 turned_knurl to numeric_parameter (as diametral_pitch)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'diametral pitch'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
plane_angle_measure_with_unit}

5.1.5.85.2 turned_knurl to numeric_parameter (as number_of_teeth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'number of teeth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.5.85.3 turned_knurl to numeric_parameter (as major_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'major diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.85.4 turned_knurl to numeric_parameter (as nominal_diameter)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'nominal diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.85.5 turned_knurl to numeric_parameter (as root_fillet)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'root fillet'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.5.85.6 turned_knurl to numeric_parameter (as tooth_depth)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: turned_knurl <=
feature_definition <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'tooth depth'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.6 manufacturing_part_properties UoF

5.1.6.1 Alternate_material

AIM element: product_definition
Source: ISO 10303-41
Rules: material_is_specified_for_part – (See 5.2.4.18)

5.1.6.1.1 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

5.1.6.1.2 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

5.1.6.2 Descriptive_parameter

AIM element: descriptive_representation_item
Source: ISO 10303-45

5.1.6.2.1 descriptive_string

AIM element: descriptive_representation_item.description
Source: ISO 10303-45

5.1.6.3 Hardness

AIM element: material_property_representation
Source: ISO 10303-45
Reference Path: {material_property_representation
material_property_representation.dependent_environment->
data_environment}

5.1.6.3.1 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[1] ->
{representation_item
representation_item.name = 'high value'}
{representation_item
representation_item=>
qualified_representation_item
qualified_representation_item.qualifiers->
type_qualifier
type_qualifier.name='high value'}
representation_item =>
measure_representation_item

5.1.6.3.2 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[1] ->
{representation_item
representation_item.name = 'low value'}
{representation_item
representation_item=>
qualified_representation_item
qualified_representation_item.qualifiers->
type_qualifier
type_qualifier.name='low value'}
representation_item =>
measure_representation_item

5.1.6.3.3 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[1] ->
{representation_item
representation_item.name = 'nominal'}
{representation_item
representation_item=>
qualified_representation_item
qualified_representation_item.qualifiers->
type_qualifier
type_qualifier.name='nominal'}
representation_item =>
measure_representation_item

5.1.6.3.4 scale

AIM element: data_environment.name
Source: ISO 10303-522
Reference Path: material_property_representation
material_property_representation.dependent_environment->
data_environment
data_environment.name = 'hardness'

5.1.6.4 Material

AIM element: product_definition
Source: ISO 10303-41

5.1.6.4.1 material_description

AIM element: product_definition.description
Source: ISO 10303-41

5.1.6.4.2 material_id

AIM element: material_designation.name
Source: ISO 10303-45
Reference Path: product_definition
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
material_designation.definitions[i]
material_designation
material_designation.name

5.1.6.4.3 stock_size

AIM element: product_definition.id
Source: ISO 10303-41

5.1.6.4.4 material to material_property (as material_characteristics)

AIM element: PATH
Reference Path: product_definition
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition=>
material_property

5.1.6.4.5 material to specification (as material_specification)

AIM element: PATH
Reference Path: product_definition =>
product_definition_with_associated_documents
product_definition_with_associated_documents.documentation_ids[i] ->
document

5.1.6.5 Material_property

AIM element: material_property
Source: ISO 10303-45

5.1.6.5.1 material_property to hardness (as material_hardness)

AIM element: PATH
Reference Path: material_property <=
property_definition <-
property_definition_representation.definition
property_definition_representation

5.1.6.5.2 material_property to property_parameter (as property_-characteristic)

AIM element: PATH
Reference Path: material_property <=
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
representation_item

5.1.6.6 Numeric_parameter

AIM element: measure_representation_item
Source: ISO 10303-45

5.1.6.6.1 parameter_units

AIM element: named_unit
Source: ISO 10303-41
Rules: dependent_instantiable_named_unit – (See 5.2.4.9)
Reference Path: measure_representation_item <=
measure_with_unit
measure_with_unit.unit_component ->
unit
unit = named_unit
named_unit

5.1.6.6.2 parameter_value

AIM element: measure_value
Source: ISO 10303-41
Reference Path: measure_representation_item <=
measure_with_unit
measure_with_unit.value_component ->
measure_value

5.1.6.7 Numeric_parameter_with_tolerance

AIM element: [measure_representation_item]
[qualified_representation_item]
Source: ISO 10303-45

5.1.6.7.1 numeric_parameter_with_tolerance to limits_and_fits (as implicit_-tolerance)

AIM element: IDENTICAL MAPPING

5.1.6.7.2 numeric_parameter_with_tolerance to plus_minus_value (as implicit_tolerance)

AIM element: IDENTICAL MAPPING

5.1.6.7.3 numeric_parameter_with_tolerance to tolerance_limit (as implicit_-tolerance)

AIM element: PATH
Reference Path: qualified_representation_item
qualified_representation_item.qualifiers[i] ->
value_qualifier
value_qualifier = type_qualifier
type_qualifier

5.1.6.7.4 numeric_parameter_with_tolerance to tolerance_range (as implicit_tolerance)

AIM element: IDENTICAL MAPPING

5.1.6.8 Part_property

AIM element: property_definition
Source: ISO 10303-41
Reference Path: {property_definition
property_definition.name = 'part property'}

5.1.6.8.1 part_property to property_parameter (as property_characteristic)

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

5.1.6.9 Process_property

AIM element: property_definition
Source: ISO 10303-41
Reference Path: {property_definition
property_definition.name = 'process property'}

5.1.6.9.1 process_name

AIM element: representation.name
Source: ISO 10303-43
Reference Path: property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.name

5.1.6.9.2 process_property to property_parameter (as property_-characteristic)

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

5.1.6.10 Property

AIM element: property_definition
Source: ISO 10303-41

5.1.6.10.1 property_name

AIM element: property_definition.name
Source: ISO 10303-41

5.1.6.10.2 property to material_property (as material_characteristic)

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

5.1.6.10.3 property to part_property (as part_property_characteristic)

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'}

5.1.6.10.4 property to process_property (as process_characteristic)

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'}

5.1.6.10.5 property to shape_aspect (as property_characteristic)

AIM element: PATH
Reference Path: property_definition
property_definition.definition ->
characterized_definition
characterized_definition = shape_definition
shape_definition
shape_definition = shape_aspect
shape_aspect

5.1.6.10.6 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

5.1.6.10.7 property to surface_property (as surface_characteristic)

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'}

5.1.6.11 Property_parameter

AIM element: representation_item
Source: ISO 10303-43

5.1.6.11.1 parameter_name

AIM element: representation_item.name
Source: ISO 10303-43

5.1.6.12 Surface_property

AIM element: property_definition
Source: ISO 10303-41
Reference Path: {property_definition
property_definition.name = 'surface property'}

5.1.6.12.1 surface_finish

AIM element: property_definition
Source: ISO 10303-41
Reference Path: {property_definition
property_definition.description = 'surface finish'}

5.1.6.12.2 surface_property to property_parameter (as property_-characteristic)

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

5.1.7 manufacturing_process_control_documentation UoF

5.1.7.1 Customer_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'customer order'}

5.1.7.1.1 delivery_date

AIM element: date
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action
feature_based_pp_dated_item = directed_action
feature_based_pp_dated_item <-
feature_based_pp_date_assignment.items[i]
feature_based_pp_date_assignment <=
{date_assignment
date_assignment.role ->
date_role
date_role.name = 'delivery date'}
date_assignment
date_assignment.assigned_date ->
date

5.1.7.1.2 material_disposition

AIM element: action.description
Source: ISO 10303-41
Reference Path: directed_action <=
executed_action <=
action
action.description

5.1.7.1.3 order_number

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.7.1.4 order_status

AIM element: action_status
Source: ISO 10303-41
Reference Path: directed_action <=
executed_action <=
action <-
action_status.assigned_action
action_status

5.1.7.1.5 special_instructions

AIM element: action_directive.comment
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.comment

5.1.7.1.6 customer_order to ordered_part (as quantity_ordered)

AIM element: PATH
Reference Path: directed_action <=
executed_action <=
action <-
action_assignment.assigned_action
action_assignment =>
ordered_part

5.1.7.1.7 customer_order to person_in_organization (as customer)

AIM element: PATH
Reference Path: directed_action
directed_action.directive ->
action_directive
feature_based_pp_person_and_organization_item = action_directive
feature_based_pp_person_and_organization_item <-
feature_based_pp_person_and_organization_assignment.items[i]
feature_based_pp_person_and_organization_assignment <=
person_and_organization_assignment
person_and_organization_assignment.assigned_person_and_organization ->
person_and_organization

5.1.7.1.8 customer_order to project_order (as initiated_order)

AIM element: PATH
Source: ISO 10303-522
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action <=
executed_action <=
action <-
action_relationship.relating_action
action_relationship
action_relationship.related_action ->
{action
action.name = 'project order'}
action =>
executed_action =>
directed_action

5.1.7.2 Digital_technical_data_package_work_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'digital technical data package work order'}

5.1.7.2.1 order_id

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.7.3 Ordered_part

AIM element: ordered_part
Source: ISO 10303-225
Rules: part_requires_project_order -- (See 5.2.4.23)
subtype_mandatory_characterized_object – (See 5.2.4.36)
Reference Path: ordered_part <=
[action_assignment]
[characterized_object]

5.1.7.3.1 quantity_required

AIM element: measure_with_unit.value_component
Source: ISO 10303-41
Reference Path: ordered_part <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
representation_item =>
measure_representation_item <=
measure_with_unit
measure_with_unit.value_component
{measure_with_unit.value_component ->
measure_value
measure_value = count_measure
count_measure}

5.1.7.3.2 quantity_unit_of_measure

AIM element: measure_with_unit.unit_component
Source: ISO 10303-41
Reference Path: ordered_part <=
characterized_object
characterized_definition = characterized_object
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
representation_item =>
measure_representation_item <=
measure_with_unit
measure_with_unit.unit_component

5.1.7.4 Pedigree_creation_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'pedigree creation order'}

5.1.7.4.1 order_id

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.7.5 Project_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
part_requires_project_order -- (See 5.2.4.23)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'project order'}

5.1.7.5.1 project_order_id

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.7.5.2 project_order to approval (as release_authorization)

AIM element: PATH
Rules: project_order_requires_approval – (See 5.2.4.27)
Reference Path: directed_action
feature_based_pp_approved_item = directed_action
feature_based_pp_approved_item <-
feature_based_pp_approval_assignment.items[i]
feature_based_pp_approval_assignment <=
approval_assignment
approval_assignment.assigned_approval ->
approval

5.1.7.5.3 project_order to digital_technical_data_packagework_order (as technical_data_package_status)

AIM element: PATH
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action <=
executed_action <=
action <-
action_relationship.relating_action
action_relationship
action_relationship.related_action ->
{action
action.name = 'digital technical data package work order'}
action =>
executed_action =>
directed_action

5.1.7.5.4 project_order to pedigree_creation_order (as pedigree_creation_-status)

AIM element: PATH
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action <=
executed_action <=
action <-
action_relationship.relating_action
action_relationship
action_relationship.related_action ->
{action
action.name = 'pedigree creation order'}
action =>
executed_action =>
directed_action

5.1.7.5.5 project_order to part (as part_status)

AIM element: PATH
Rules: part_requires_project_order -- (See 5.2.4.23)
Reference Path: directed_action <=
executed_action <=
action <-
action_assignment.assigned_action
action_assignment =>
feature_based_pp_action_assignment
feature_based_pp_action_assignment.items[i] ->
feature_based_pp_action_item
feature_based_pp_action_item = product_definition_formation
product_definition_formation

5.1.7.5.6 project_order to requisition (as ordered_resource)

AIM element: PATH
Reference Path: directed_action
feature_based_pp_document_item = directed_action
feature_based_pp_document_item <-
feature_based_pp_document_reference.items[i]
feature_based_pp_document_reference <=
document_reference
document_reference.assigned_document ->
document

5.1.7.5.7 project_order to resource_acquisition_order (as resource_-acquisition_status)

AIM element: PATH
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action <=
executed_action <=
action <-
action_relationship.relating_action
action_relationship
action_relationship.related_action ->
{action
action.name = 'resource acquisition order'}
action =>
executed_action =>
directed_action

5.1.7.5.8 project_order to shop_work_order (as shop_work_status)

AIM element: PATH
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: directed_action <=
executed_action <=
action <-
action_relationship.relating_action
action_relationship
action_relationship.related_action ->
{action
action.name = 'shop work order'}
action =>
executed_action =>
directed_action

5.1.7.6 Resource_acquisition_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'resource acquisition order'}

5.1.7.6.1 order_id

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.7.7 Shop_work_order

AIM element: directed_action
Source: ISO 10303-41
Rules: project_order_tracking_relationships – (See 5.2.4.28)
Reference Path: {directed_action <=
executed_action <=
action
action.name = 'shop work order'}

5.1.7.7.1 order_id

AIM element: action_directive.name
Source: ISO 10303-41
Reference Path: directed_action
directed_action.directive ->
action_directive
action_directive.name

5.1.8 manufacturing_process_requirement_documents UoF

5.1.8.1 Part_dimensioning_standard

AIM element: applied_document_reference
Source: ISO 10303-224
Reference Path: applied_document_reference<=
document_reference
{document_reference.role->
object_role
object_role.name= ’dimensioning standard’}

5.1.8.1.1 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

5.1.8.2 Specification

AIM element: document
Source: ISO 10303-41

5.1.8.2.1 release_date

AIM element: date
Source: ISO 10303-41
Reference Path: document=>
feature_based_pp_dated_item = document
 feature_based_pp_dated_item <-
feature_based_pp_date_assignment.items[i]
 feature_based_pp_date_assignment <=
{date_assignment
date_assignment.role ->
date_role
date_role.name = 'release date'}
date_assignment
date_assignment.assigned_date ->
date

5.1.8.2.2 specification_class

AIM element: document_with_class.class
Source: ISO 10303-41
Reference Path: document=>
document_with_class

5.1.8.2.3 specification_description

AIM element: document.description
Source: ISO 10303-41
Reference Path: document
document.description

5.1.8.2.4 specification_id

AIM element: document.id
Source: ISO 10303-41
Reference Path: document
document.id

5.1.8.2.5 version_id

AIM element: identification_assignment.assigned_id
Source: ISO 10303-41
Reference Path: document
identification_item = document
identification_item <-
applied_identification_assignment.items[i]
applied_identification_assignment <=
identification_assignment
{identification_assignment.role ->
identification_role
identification_role.name = 'version'}
identification_assignment.assigned_id

5.1.8.2.6 specification to specification_usage_constraint (as constraint)

AIM element: PATH
Reference Path: document <-
document_usage_constraint.source
document_usage_constraint

5.1.8.3 Specification_usage_constraint

AIM element: document_usage_constraint
Source: ISO 10303-41

5.1.8.3.1 class_id

AIM element: document_usage_constraint.subject_element_value
Source: ISO 10303-41

5.1.8.3.2 element

AIM element: document_usage_constraint.subject_element
Source: ISO 10303-41

5.1.9 measurement_limitations UoF

5.1.9.1 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: shape_representation_subtype_exclusiveness – (See 5.2.4.35)

5.1.9.1.1 angular_dimension_tolerance to orientation (as plane_and_-direction)

#1: if directed = true

AIM element: PATH
Rule: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)
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

5.1.9.2 Angular_size_dimension_tolerance

AIM element: angular_size
Source: ISO 10303-47

5.1.9.2.1 full_or_half

AIM element: descriptive_representation_item.description
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
descriptive_representation_item
descriptive_representation_item.description
{(descriptive_representation_item.description= 'full angle')
(descriptive_representation_item.description= 'half angle')}

5.1.9.2.2 major_angle

AIM element: angular_size.angle_selection
Source: ISO 10303-47

5.1.9.3 Angularity_tolerance

AIM element: angularity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)

5.1.9.3.1 segment_size

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: angularity_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance =>
geometric_tolerance_with_defined_unit
geometric_tolerance_with_defined_unit.unit_size ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.3.2 angularity_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: angularity_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.4 Circular_runout_tolerance

AIM element: circular_runout_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.4.1 runout_angle

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: circular_runout_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition =>
runout_zone_definition
runout_zone_definition.orientation ->
runout_zone_orientation
runout_zone_orientation.angle ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.4.2 circular_runout_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: circular_runout_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.5 Circularity_tolerance

AIM element: roundness_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.6 Compound_datum

AIM element: common_datum
(see NOTE)
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)

NOTE The datum that represents a compound_datum uses at least four shape_aspect_relationships. At least two shape_aspect_relationships reference the datum associated to the compound_datum. Each datum referenced by the compound_datum has ashape_aspect_relationship to a datum_feature. The compound_datum has a shape_aspect_relationship that references the same datum_features.

5.1.9.6.1 compound_datum to datum_feature (as element)

AIM element: PATH
Reference Path: common_datum
datum <=
shape_aspect <-
shape_aspect_relationship.relating _shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect =>
datum

5.1.9.7 Concentricity_tolerance

AIM element: concentricity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)

5.1.9.7.1 value_qualifier

AIM element: tolerance_zone_form.name
Source: ISO 10303-47
Reference Path: concentricity_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone
tolerance_zone.form ->
tolerance_zone_form
tolerance_zone_form.name

5.1.9.7.2 concentricity_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: concentricity_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.8 Curved_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference Path: {dimensional_size
dimensional_size.name = 'curve length'}

5.1.9.9 Cylindricity_tolerance

AIM element: cylindricity_tolerance
Source: ISO 10303-519

5.1.9.10 Datum

AIM element: datum_reference
Source: ISO 10303-47

5.1.9.10.1 name

AIM element: shape_aspect.name
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)
Reference Path: datum_reference
datum_reference.referenced_datum ->
datum <=
shape_aspect
shape_aspect.name

5.1.9.10.2 precedence

AIM element: datum_reference.precedence
Source: ISO 10303-47

5.1.9.11 Datum_feature

AIM element: (datum)
([datum]
[datum_feature])
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: {(datum)
(datum <=
shape_aspect <-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
datum_feature)}

5.1.9.11.1 datum_feature to datum_target_set (as datum_representation)

AIM element: PATH
Reference Path: datum
datum.established_by_relationships

5.1.9.11.2 datum_feature to material_condition_modifier (as modifier)

AIM element: PATH
Reference Path: datum <-
datum_reference.referenced_datum
datum_reference =>
referenced_modified_datum
referenced_modified_datum.modifier ->
limit_condition

5.1.9.11.3 datum_feature to shape_element (as datum_representation)

AIM element: IDENTICAL MAPPING

5.1.9.12 Datum_target

AIM element: datum_target
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34

5.1.9.12.1 identifier

AIM element: datum_target.target_id
Source: ISO 10303-47

5.1.9.13 Datum_target_set

AIM element: datum.established_by_relationships
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: {datum.established_by_relationships[i] ->
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
datum_target}

5.1.9.13.1 rule_description

AIM element: shape_aspect_relationship.description
Source: ISO 10303-41
Reference Path: datum.established_by_relationships[i] ->
shape_aspect_relationship
shape_aspect_relationship.description

5.1.9.13.2 datum_target_set to datum_target (as target_shape)

AIM element: PATH
Reference Path: datum.established_by_relationships
datum <=
shape_aspect<-
shape_aspect_relationship.related_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect =>
datum_target

5.1.9.14 Diameter_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference Path: {dimensional_size
dimensional_size.name = 'diameter'}

5.1.9.15 Dimensional_tolerance

AIM element: shape_dimension_representation
Source: ISO 10303-47

5.1.9.15.1 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)

5.1.9.15.2 dimension_note

AIM element: descriptive_representation_item.description
Source: ISO 10303-43
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33
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’)

5.1.9.15.3 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

5.1.9.15.4 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

5.1.9.15.5 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

5.1.9.15.6 dimensional_tolerance to tolerance_value (as limit)

AIM element: PATH
Reference Path: shape_dimension_representation <-
dimensional_characteristic_representation.representation
dimensional_characteristic_representation

5.1.9.16 Distance_along_curve_tolerance

AIM element: dimensional_location_with_path
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33

5.1.9.16.1 distance_along_curve_tolerance to shape_aspect (as path)

AIM element: PATH
Reference Path: dimensional_location_with_path
dimensional_location_with_path.path ->
shape_aspect

5.1.9.17 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]

5.1.9.17.1 tolerance_class

AIM element: external_source.discription
Source: ISO 10303-224
Reference Path: [externally_defined_dimension_definition<=
externally_defined_item
externally_defined_item.source ->
external_source
external_source.description

5.1.9.17.2 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'

5.1.9.17.3 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

5.1.9.18 Flatness_tolerance

#1: if segment size is specified

AIM element: (geometric_tolerance_with_defined_unit)
Source: ISO 10303-519
Reference Path: flatness_tolerance<=
{(geometric_tolerance_with_defined_unit <=
geometric_tolerance)}

#2: if segment_size is not specified

AIM element: (geometric_tolerance)
Source: ISO 10303-519
Reference Path: flatness_tolerance<=
{ (geometric_tolerance)}

5.1.9.18.1 segment_size

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: #1: geometric_tolerance_with_defined_unit
geometric_tolerance_with_defined_unit.unit_size ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.19 Geometric_tolerance

AIM element: geometric_tolerance
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33

5.1.9.19.1 geometric_tolerance_value

AIM element: measure_with_unit.value_component
Source: ISO 10303-41
Reference Path: geometric_tolerance
geometric_tolerance.magnitude ->
measure_with_unit
measure_with_unit.value_component

5.1.9.19.2 significant_digits

AIM element: precision_qualifier.precision_value
Source: ISO 10303-45
Reference Path: geometric_tolerance
geometric_tolerance.magnitude ->
measure_with_unit <-
measure_qualification.qualifiers[1] ->
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

5.1.9.19.3 unit_of_measure

AIM element: unit
Source: ISO 10303-41
Reference Path: geometric_tolerance
geometric_tolerance.magnitude ->
measure_with_unit
measure_with_unit.unit_component ->
unit

5.1.9.19.4 geometric_tolerance to material_condition_modifier (as modifier_-control)

AIM element: PATH
Reference Path: geometric_tolerance =>
modified_geometric_tolerance
modified_geometric_tolerance.modifier ->
limit_condition

5.1.9.19.5 geometric_tolerance to shape_aspect (as applied_shape)

AIM element: PATH
Reference Path: geometric_tolerance
geometric_tolerance.toleranced_shape_aspect ->
shape_aspect

5.1.9.19.6 geometric_tolerance to tolerance_zone (as zone_definition)

AIM element: PATH
Reference Path: geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone

5.1.9.20 Geometric_tolerance_precedence_relationship

AIM element: geometric_tolerance_relationship
Source: ISO 10303-47
Reference Path: {geometric_tolerance_relationship
geometric_tolerance_relationship.name = 'precedence'}

5.1.9.20.1 geometric_tolerance_precedence_relationship to geometric_-tolerance(as base_shape_tolerance)

AIM element: PATH
Reference Path: geometric_tolerance_precedence_relationship <=
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect <-
geometric_tolerance.toleranced_shape_aspect
geometric_tolerance

5.1.9.20.2 geometric_tolerance_precedence_relationship to geometric_-tolerance (as pattern_shape_tolerance)

AIM element: PATH
Reference Path: geometric_tolerance_precedence_relationship <=
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect <-
geometric_tolerance.toleranced_shape_aspect
geometric_tolerance

5.1.9.21 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'

5.1.9.21.1 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

5.1.9.22 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'

5.1.9.22.1 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

5.1.9.23 Limits_and_fits

AIM element: limits_and_fits
Source: ISO 10303-47

5.1.9.23.1 deviation

AIM element: limits_and_fits.form_variance
Source: ISO 10303-47

5.1.9.23.2 fitting_type

AIM element: limits_and_fits.zone_variance
Source: ISO 10303-47

5.1.9.23.3 grade

AIM element: limits_and_fits.grade
Source: ISO 10303-47

5.1.9.24 Linear_profile_tolerance

AIM element: line_profile_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.24.1 linear_profile_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: line_profile_tolerance <=
geometric_tolerance =>
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.24.2 linear_profile_tolerance to orientation (as affected_plane)

AIM element: PATH
Reference Path: line_profile_tolerance <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.25 Location_dimension_tolerance

#1: if directed = true

AIM element: (directed_dimensional_location)
Source: ISO 10303-224
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33
Reference Path: #1: directed_dimensional_location <=
dimensional_location

#2: if directed = false

AIM element: (dimensional_location)
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33

5.1.9.25.1 location_dimension_tolerance to orientation (as plane_and_-direction)

#1: if directed = true

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33
Reference Path: (directed_dimensional_location <=
dimensional_location <=)

#2: if directed = false

AIM element: PATH
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33
Reference Path: (dimensional_location <=)
shape_aspect_relationship
shape_definition = shape_aspect_relationship
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.26 Location_tolerance

AIM element: dimensional_location
Source: ISO 10303-47
Rules: shape_aspect_relationship_subtype_exclusiveness – (See 5.2.4.33)5.2.4.33

5.1.9.26.1 directed

AIM element: IDENTICAL MAPPING

5.1.9.26.2 location_tolerance to shape_element (as origin_shape)

AIM element: PATH
Reference Path: dimensional_location <=
shape_aspect_relationship
shape_aspect_relationship.relating_shape_aspect ->
shape_aspect

5.1.9.26.3 location_tolerance to shape_element (as termination_shape)

AIM element: PATH
Reference Path: dimensional_location <=
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect ->
shape_aspect

5.1.9.27 Material_condition_modifier

AIM element: limit_condition
Source: ISO 10303-47

5.1.9.27.1 material_type

AIM element: IDENTICAL_MAPPING

5.1.9.28 Parallelism_tolerance

AIM element: parallelism_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.28.1 segment_size

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: parallelism_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance =>
geometric_tolerance_with_defined_unit
geometric_tolerance_with_defined_unit.unit_size ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.28.2 parallelism_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: parallelism_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.28.3 parallelism_tolerance to orientation (as affected_plane)

AIM element: PATH
Reference Path: parallelism_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.29 Perpendicularity_tolerance

AIM element: perpendicularity_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.29.1 segment_size

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: perpendicularity_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance =>
geometric_tolerance_with_defined_unit
geometric_tolerance_with_defined_unit.unit_size ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.29.2 perpendicularity_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: perpendicularity_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.29.3 perpendicularity_tolerance to orientation (as affected_plane)

AIM element: PATH
Reference Path: perpendicularity_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.30 Placed_target

AIM element: placed_datum_target_feature
Source: ISO 10303-224
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
(shape_aspect.description = 'point')
(shape_aspect.description = 'line')
(shape_aspect.description = 'rectangle')
(shape_aspect.description = 'circle')}

5.1.9.30.1 placed_target to orientation (as placement)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: placed_datum_target_feature <=
datum_target <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.31 Plus_minus_value

#1:if the plus_minus_value is selected for tolerance_value

AIM element: #1: (tolerance_value)
Source: ISO 10303-47

#2: if the plus_minus_value is selected for numeric_parameter

AIM element: (qualified_representation_item)
Source: ISO 10303-45

5.1.9.31.1 lower_limit

#1:if the plus_minus_value is selected for tolerance_value

AIM element: #1: (tolerance_value.lower_bound)
Source: ISO 10303-47

#2: if the plus_minus_value is selected for numeric_parameter

AIM element: (standard_uncertainty.uncertainty_value)
Source: ISO 10303-47
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

5.1.9.31.2 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: #1: (tolerance_value
[tolerance_value.upper_bound ->]
[tolerance_value.lower_bound ->]
measure_with_unit <-
measure_qualification.qualifiers[1] ->)
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

#2: if the plus_minus_value is selected for numeric_parameter

AIM element: precision_qualifier.precision_value
Source: ISO 10303-45
Reference Path: (qualified_representation_item
qualified_representation_item.qualifiers[i] ->)
value_qualifier
value_qualifier = precision_qualifier
precision_qualifier
precision_qualifier.precision_value

5.1.9.31.3 upper_limit

#1:if the plus_minus_value is selected for tolerance_valuE

AIM element: (tolerance_value.upper_bound)
Source: ISO 10303-47

#2: if the plus_minus_value is selected for numeric_parameter

AIM element: (standard_uncertainty.uncertainty_value)
Source: ISO 10303-47
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

5.1.9.32 Position_tolerance

AIM element: position_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.32.1 value_qualifier

AIM element: tolerance_zone_form.name
Source: ISO 10303-47
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16
Reference Path: position_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone
tolerance_zone.form ->
tolerance_zone_form
tolerance_zone_form.name

5.1.9.32.2 position_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: position_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.32.3 position_tolerance to orientation (as affected_plane)

AIM element: PATH
Reference Path: position_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.33 Projection

AIM element: projected_zone_definition
Source: ISO 10303-47

5.1.9.33.1 projection_length

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: projected_zone_definition
projected_zone_definition.projected_length ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.33.2 projection to shape_element (as projection_end)

AIM element: PATH
Reference Path: projected_zone_definition
projected_zone_definition.projection_end ->
shape_aspect

5.1.9.34 Radial_dimension_tolerance

AIM element: dimensional_size
Source: ISO 10303-47
Reference Path: {dimensional_size
dimensional_size.name = 'radius'}

5.1.9.35 Size_tolerance

AIM element: dimensional_size
Source: ISO 10303-47

5.1.9.35.1 envelope

AIM element: representation.name
Source: ISO 10303-43
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation <=
shape_representation <=
representation
representation.name
{representation.name = 'envelope tolerance'}

5.1.9.35.2 size_tolerance to shape_element (as applied_shape)

AIM element: PATH
Reference Path: dimensional_size
dimensional_size.applies_to ->
shape_aspect

5.1.9.36 Straightness_tolerance

#1: if segment size is specified

AIM element: (geometric_tolerance_with_defined_unit)
Source: ISO 10303-519
Reference Path: straightness_tolerance<=
(geometric_tolerance_with_defined_unit <=
geometric_tolerance)

#2: if segment size is not specified

AIM element: (geometric_tolerance)
Source: ISO 10303-519
Reference Path: straightness_tolerance<=
(geometric_tolerance)}

5.1.9.36.1 segment_size

#1: if segment size is specified

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Reference Path: geometric_tolerance_with_defined_unit
geometric_tolerance_with_defined_unit.unit_size ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.36.2 straightness_tolerance to orientation (as affected_plane)

#1: if segment size is specified

AIM element: PATH
Reference Path: (geometric_tolerance_with_defined_unit <=
geometric_tolerance <-)
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

#2: if segment size is not specified

AIM element: PATH
Reference Path: (geometric_tolerance <-)
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.37 Surface_profile_tolerance

AIM element: surface_profile_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.37.1 surface_profile_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: surface_profile_tolerance <=
geometric_tolerance =>
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.38 Symmetry_tolerance

AIM element: symmetry_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.38.1 symmetry_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: symmetry_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.38.2 symmetry_tolerance to orientation (as affected_plane)

AIM element: PATH
Reference Path: symmetry_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <=
shape_aspect <-
shape_aspect_relationship.relating_shape_aspect
shape_aspect_relationship
shape_aspect_relationship.related_shape_aspect->
shape_aspect
{shape_aspect
shape_aspect.description = 'affected plane'}
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
{representation_item
representation_item.name = 'affected plane'}
representation_item =>
geometric_representation_item =>
placement

5.1.9.39 Target_area

AIM element: datum_target
Source: ISO 10303-47
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34

5.1.9.39.1 target_area to shape_element (as area_shape)

AIM element: IDENTICAL MAPPING

5.1.9.40 Target_circle

AIM element: placed_datum_target_feature
Source: ISO 10303-224
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'circle'}

5.1.9.40.1 target_diameter

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: placed_datum_target_feature <=
datum_target <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'target diameter'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.9.41 Target_line

AIM element: placed_datum_target_feature
Source: ISO 10303-224
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'line'}

5.1.9.41.1 target_length

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: placed_datum_target_feature <=
datum_target <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'target length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.9.42 Target_point

AIM element: placed_datum_target_feature
Source: ISO 10303-224
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'point'}

5.1.9.43 Target_rectangle

AIM element: placed_datum_target_feature
Source: ISO 10303-224
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: placed_datum_target_feature <=
datum_target
{datum_target <=
shape_aspect
shape_aspect.description = 'rectangle'}

5.1.9.43.1 target_length

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: placed_datum_target_feature <=
datum_target <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'target length'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.9.43.2 target_width

AIM element: measure_representation_item
Source: ISO 10303-45
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: placed_datum_target_feature <=
datum_target <=
shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
{representation =>
shape_representation =>
shape_representation_with_parameters}
representation
representation.items[i] ->
{representation_item
representation_item.name = 'target width'}
representation_item =>
measure_representation_item
{measure_representation_item <=
measure_with_unit =>
length_measure_with_unit}

5.1.9.44 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'

5.1.9.44.1 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

5.1.9.45 Tolerance_limit

AIM element: type_qualifier
Source: ISO 10303-45

5.1.9.45.1 limit_qualifier

AIM element: type_qualifier.name
Source: ISO 10303-45
Reference Path: (type_qualifier.name='maximum')
(type_qualifier.name='minimum')
(type_qualifier.name)

5.1.9.46 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}

5.1.9.46.1 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

5.1.9.46.2 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

5.1.9.46.3 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

5.1.9.47 Tolerance_value

AIM element: dimensional_characteristic_representation
Source: ISO 10303-45

5.1.9.47.1 tolerance_value to limits_and_fits (as defined_value)

AIM element: PATH
Source: ISO 10303-522
Reference Path: dimensional_characteristic_representation
dimensional_characteristic_representation.dimension ->
dimensional_characteristic <-
plus_minus_tolerance.toleranced_dimension
plus_minus_tolerance
plus_minus_tolerance.range ->
tolerance_method_definition
tolerance_method_definition = limits_and_fits
limits_and_fits

5.1.9.47.2 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

5.1.9.47.3 tolerance_value to tolerance_limit (as defined_value)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
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

5.1.9.47.4 tolerance_value to tolerance_range (as defined_value)

AIM element: PATH
Reference Path: dimensional_characteristic_representation
dimensional_characteristic_representation.representation ->
shape_dimension_representation

5.1.9.48 Tolerance_zone

AIM element: tolerance_zone
Source: ISO 10303-47

5.1.9.48.1 common_zone

AIM element: tolerance_zone.defining_tolerance
Source: ISO 10303-47

5.1.9.48.2 form_type

AIM element: tolerance_zone_form.name
Source: ISO 10303-47
Reference Path: tolerance_zone
tolerance_zone.form ->
tolerance_zone_form
tolerance_zone_form.name

5.1.9.48.3 tolerance_zone to projection (as extended_shape)

AIM element: PATH
Reference Path: tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition =>
projected_zone_definition

5.1.9.48.4 tolerance_zone to tolerance_zone_definition (as zone_definition)

AIM element: PATH
Reference Path: tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition

5.1.9.49 Tolerance_zone_definition

AIM element: tolerance_zone_definition
Source: ISO 10303-47

5.1.9.49.1 tolerance_zone_definition to shape_element (as first_element)

AIM element: PATH
Reference Path: tolerance_zone_definition
tolerance_zone_definition.boundaries[i] ->
shape_aspect

5.1.9.49.2 tolerance_zone_definition to shape_element (as second_element)

AIM element: PATH
Reference Path: tolerance_zone_definition
tolerance_zone_definition.boundaries[i] ->
shape_aspect

5.1.9.50 Total_runout_tolerance

AIM element: total_runout_tolerance
Source: ISO 10303-519
Rules: geometric_tolerance_subtype_exclusiveness – (See 5.2.4.16)5.2.4.16

5.1.9.50.1 runout

AIM element: [measure_with_unit.value_component]
[measure_with_unit.unit_component]
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: total_runout_tolerance <=
geometric_tolerance_with_datum_reference <=
geometric_tolerance <-
tolerance_zone.defining_tolerance[i]
tolerance_zone <-
tolerance_zone_definition.zone
tolerance_zone_definition =>
runout_zone_definition
runout_zone_definition.orientation ->
runout_zone_orientation
runout_zone_orientation.angle ->
measure_with_unit
[measure_with_unit.value_component]
[measure_with_unit.unit_component]

5.1.9.50.2 total_runout_tolerance to datum (as geometric_reference)

AIM element: PATH
Reference Path: total_runout_tolerance <=
geometric_tolerance_with_datum_reference
geometric_tolerance_with_datum_reference.datum_system[i] ->
datum_reference

5.1.9.51 Width_dimension

#1:if the width_dimension is specified and the optional path is specified

AIM element: dimensional_size_with_path
Source: ISO 10303-47
Reference Path: dimensional_size_with_path<=
dimensional_size
dimensional_size.name='width'

#2:if the width_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='width'

5.1.9.51.1 width_dimensionto shape_aspect (as path)

#1:if the width_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 width_dimension is specified and the optional path is not specified

AIM element: dimensional_size
Source: ISO 10303-47

5.1.10 part_administration_data UoF

5.1.10.1 Approval

AIM element: approval
Source: ISO 10303-41
Rules: approval_requires_approval_date_time – (See 5.2.4.2)
approval_requires_approval_person_organization – (See 5.2.4.3)
dependent_instantiable_approval_status – (See 5.2.4.7)
part_to_approval – (See 5.2.4.24)
project_order_requires_approval – (See 5.2.4.27)
restrict_approval_status – (See 5.2.4.30)

5.1.10.1.1 approval_date

AIM element: date
Source: ISO 10303-41
Rules: approval_requires_approval_date_time – (See 5.2.4.2)
dependent_instantiable_date – (See 5.2.4.8)
Reference Path: approval <-
approval_date_time.dated_approval
approval_date_time
approval_date_time.date_time ->
date_time_select
date_time_select = date
date

5.1.10.1.2 status

AIM element: approval_status.name
Source: ISO 10303-41
Rules: dependent_instantiable_approval_status – (See 5.2.4.7)
restrict_approval_status – (See 5.2.4.30)
Reference Path: approval
approval.status ->
approval_status
approval_status.name
{(approval_status.name = 'approved')
(approval_status.name = 'not yet approved')
(approval_status.name = 'disapproved')
(approval_status.name = 'withdrawn')}

5.1.10.1.3 approval to person_in_organization (as approval_authority)

AIM element: PATH
Rules: approval_requires_approval_person_organization – (See 5.2.4.3)
Reference Path: approval <-
approval_person_organization.authorized_approval
approval_person_organization
approval_person_organization.person_organization ->
person_organization_select
person_organization_select = person_and_organization
person_and_organization

5.1.10.2 Organization

AIM element: organization
Source: ISO 10303-41

5.1.10.2.1 organization_address

AIM element: address
Source: ISO 10303-41
Reference Path: organization <-
organizational_address.organizations[i]
organizational_address <=
address

5.1.10.2.2 organization_id

AIM element: organization.id
Source: ISO 10303-41

5.1.10.2.3 organization_name

AIM element: organization.name
Source: ISO 10303-41

5.1.10.3 Person

AIM element: person
Source: ISO 10303-41

5.1.10.3.1 person_address

AIM element: address
Source: ISO 10303-41
Reference Path: person <-
personal_address.people[i]
personal_address <=
address

5.1.10.3.2 person_id

AIM element: person.id
Source: ISO 10303-41

5.1.10.3.3 person_name

AIM element: [person.first_name]
[person.last_name]
Source: ISO 10303-41

5.1.10.3.4 person_phone_number

AIM element: address.telephone_number
Source: ISO 10303-41
Reference Path: person <-
personal_address.people[i]
personal_address <=
address
address.telephone_number

5.1.10.4 Person_in_organization

AIM element: person_and_organization
Source: ISO 10303-41

5.1.10.4.1 role

AIM element: person_and_organization_role.name
Source: ISO 10303-41
Reference Path: person_and_organization <-
person_and_organization_assignment.assigned_person_and_organization
person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.name

5.1.10.4.2 person_in_organization to organization (as company)

AIM element: PATH
Reference Path: person_and_organization
person_and_organization.the_organization ->
organization

5.1.10.4.3 person_in_organization to person (as employee)

AIM element: PATH
Reference Path: person_and_organization
person_and_organization.the_person ->
person

5.1.11 part model UoF

5.1.11.1 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'

5.1.11.2 Manufactured_assembly_relationship

AIM element: next_assembly_usage_occurrence
Source: ISO 10303-44

5.1.11.2.1 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

5.1.11.2.2 manufactured_assembly_relationship to part (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

5.1.11.2.3 manufactured_assembly_relationship to part_placement (as orientation)

AIM element: PATH
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_map.mapped_representation
representation_map <-
mapped_item.mapping_source]
mapped_item

5.1.11.3 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'

5.1.11.3.1 mating_type

AIM element: product_definition.name
Source: ISO 10303-41

5.1.11.3.2 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'}

5.1.11.3.3 mating_definition to shape_element (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

5.1.11.3.4 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

5.1.11.3.5 mating_definition to externally_defined_representation (as standard_parts)

AIM element: PATH
Rules: mating_definition_requires_externally_defined_representation – (See 5.2.4.19)
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
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.name='standard part fastener'
representation=>
externally_defined_representation_with_parameters

5.1.11.4 Mating_definiton_relationship

AIM element: product_definiton_relationship
Source: ISO 10303-41
Reference Path: product_definiton_relationship
product_definiton_relationship.name='mating material'

5.1.11.4.1 mating_definition_relationship to mating_definition (as mating_-part_definition)

AIM element: PATH
Reference Path: product_definiton_relationship
product_definiton_relationship.relating_product_definitoin->
product_definition
(product_definition
product_definition.frame_of_reference->
product_definition_context<=
application_context_element
application_context_element.name='mating definition')

5.1.11.4.2 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

5.1.11.4.3 mating_definition_relationship to single_piece_part (as mated_-part)

AIM element: PATH
Reference Path: product_definiton_relationship
product_definiton_relationship.related_product_definitoin->
product_definition

5.1.11.5 Mating_relationship

AIM element: product_definiton_relationship
Source: ISO 10303-41
Reference Path: product_definiton_relationship
product_definiton_relationship.name='mating membership'

5.1.11.5.1 mating_relationship to single_piece_part (as predecessor)

AIM element: PATH
Reference Path: product_definiton_relationship
product_definiton_relationship.related_product_definitoin->
product_definition

5.1.11.5.2 mating_relationship to single_piece_part (as successor)

AIM element:	PATH
Reference Path:	product_definiton_relationship product_definiton_relationship.relating_product_definitoin-> product_definition

5.1.11.6 Part

AIM element:	product_definition_formation
Source:	ISO 10303-41
Rules:	part_requires_project_order – (See 5.2.4.23) part_to_approval – (See 5.2.4.24) product_definition_formation_requires_security_classification(See 5.2.4.26)
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'

5.1.11.6.1 part_description

AIM element:	product.description
Source:	ISO 10303-41
Rules:	product_requires_version – (See 5.2.4.25)
Reference Path:	product_definition_formation product_definition_formation.of_product -> product product.description

5.1.11.6.2 part_id

AIM element:	product.id
Source:	ISO 10303-41
Rules:	product_requires_version – (See 5.2.4.25)
Reference Path:	product_definition_formation product_definition_formation.of_product -> product product.id

5.1.11.6.3 part_name

AIM element: product.name
Source: ISO 10303-41
Rules: product_requires_version – (See 5.2.4.25)
Reference Path: product_definition_formation
product_definition_formation.of_product ->
product
product.name

5.1.11.6.4 part_revision_id

AIM element: product_definition_formation.id
Source: ISO 10303-41

5.1.11.6.5 security_classification

AIM element: security_classification
Source: ISO 10303-41
Rules: dependent_instantiable_security_classification_level – (See 5.2.4.13)
product_definition_formation_requires_security_classification(See 5.2.4.26)
restrict_security_classification_level – (See 5.2.4.32)
Reference Path: product_definition_formation
feature_based_pp_classified_item = product_definition_formation
feature_based_pp_classified_item <-
feature_based_pp_security_classification_assignment.items[i]
feature_based_pp_security_classification_assignment <=
security_classification_assignment
security_classification_assignment.assigned_security_classification ->
security_classification

5.1.11.6.6 part to approval (as manufacture_authorization)

AIM element: PATH
Rules: part_to_approval – (See 5.2.4.24)
Reference Path: product_definition_formation
feature_based_pp_approved_item = product_definition_formation
feature_based_pp_approved_item <-
feature_based_pp_approval_assignment.items[i]
feature_based_pp_approval_assignment <=
approval_assignment
approval_assignment.assigned_approval ->
approval

5.1.11.6.7 part to ordered_part (as quantity_ordered)

AIM element: PATH
Rules: part_requires_project_order – (See 5.2.4.23)
Reference Path: product_definition_formation
feature_based_pp_ordered_item = product_definition_formation
feature_based_pp_ordered_item <-
ordered_part.items[i]
ordered_part

5.1.11.6.8 part to organization (as manufactured_by_organization)

AIM element: PATH
Reference Path: product_definition_formation
feature_based_pp_organization_item = product_definition_formation
feature_based_pp_organization_item <-
feature_based_pp_organization_assignment.items[i]
feature_based_pp_organization_assignment <=
{organization_assignment
organization_assignment.role ->
organization_role
organization_role.name = 'manufacturer'}
organization_assignment
organization_assignment.assigned_organization ->
organization

5.1.11.6.9 part to organization (as owned_by_organization)

AIM element: PATH
Reference Path: product_definition_formation
feature_based_pp_organization_item = product_definition_formation
feature_based_pp_organization_item <-
feature_based_pp_organization_assignment.items[i]
feature_based_pp_organization_assignment <=
{organization_assignment
organization_assignment.role ->
organization_role
organization_role.name = 'owner'}
organization_assignment
organization_assignment.assigned_organization ->
organization

5.1.11.6.10 part to person_in_organization (as manufactured_by_person)

AIM element: PATH
Reference Path: product_definition_formation
feature_based_pp_person_and_organization_item = product_definition_formation
feature_based_pp_person_and_organization_item <-
feature_based_pp_person_and_organization_assignment.items[i]
feature_based_pp_person_and_organization_assignment <=
{person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.name = 'manufacturer'}
person_and_organization_assignment
person_and_organization_assignment.assigned_person_and_organization ->
person_and_organization

5.1.11.6.11 part to person_in_organization (as owned_by_person)

AIM element: PATH
Reference Path: product_definition_formation
feature_based_pp_person_and_organization_item = product_definition_formation
feature_based_pp_person_and_organization_item <-
feature_based_pp_person_and_organization_assignment.items[i]
feature_based_pp_person_and_organization_assignment <=
{person_and_organization_assignment
person_and_organization_assignment.role ->
person_and_organization_role
person_and_organization_role.name = 'owner'}
person_and_organization_assignment
person_and_organization_assignment.assigned_person_and_organization ->
person_and_organization

5.1.11.6.12 part 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

5.1.11.6.13 part to shape (as physical_form)

AIM element: PATH
Reference Path: product_definition_formation <-
product_definition.formation
product_definition
characterized_product_definition = product_definition
characterized_product_definition
characterized_definition = characterized_product_definition
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape

5.1.11.7 Single_piece_part

AIM element: product_definition_formation
Source: ISO 10303-41

5.1.11.7.1 single_piece_part to material (as material_definition)

AIM element: PATH
Rules: material_is_specified_for_part – (See 5.2.4.18)
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
material_designation}

5.1.11.7.2 single_piece_part to alternate_material (as alternate_material_-definition)

AIM element: PATH
Rules: material_is_specified_for_part – (See 5.2.4.18)
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]}

5.1.12 requisitions UoF

5.1.12.1 Cutting_tool_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'cutting tool requisition'}

5.1.12.2 Dedicated_fixture_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'dedicated fixture requisition'}

5.1.12.3 Indirect_stock_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'indirect stock requisition'}

5.1.12.4 Machine_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'machine requisition'}

5.1.12.5 Material_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'material requisition'}

5.1.12.6 Modular_fixture_requisition

AIM element: document
Source: ISO 10303-41
Reference Path: {document
document.kind ->
document_type
document_type.product_data_type = 'modular fixture requisition'}

5.1.12.7 Requisition

AIM element: document
Source: ISO 10303-41

5.1.12.7.1 quantity_ordered

AIM element: document.name
Source: ISO 10303-41

5.1.12.7.2 required_delivery_date

AIM element: date
Source: ISO 10303-41
Rules: dependent_instantiable_date – (See 5.2.4.8)
Reference Path: document=>
document_with_class
feature_based_pp_dated_item = document_with_class
feature_based_pp_dated_item <-
feature_based_pp_date_assignment.items[i]
feature_based_pp_date_assignment <=
{date_assignment
date_assignment.role ->
date_role
date_role.name = 'required delivery date'}
date_assignment
date_assignment.assigned_date ->
date

5.1.12.7.3 requisition_date

AIM element: date
Source: ISO 10303-41
Rules: dependent_instantiable_date – (See 5.2.4.8)
Reference Path: document=>
document_with_class
feature_based_pp_dated_item = document_with_class
feature_based_pp_dated_item <-
feature_based_pp_date_assignment.items[i]
feature_based_pp_date_assignment <=
{date_assignment
date_assignment.role ->
date_role
date_role.name = 'requisition date'}
date_assignment
date_assignment.assigned_date ->
date

5.1.12.7.4 requisition_description

AIM element: document.description
Source: ISO 10303-41

5.1.12.7.5 requisition_number

AIM element: document.id
Source: ISO 10303-41

5.1.13 shape_representation_for_machining UoF

5.1.13.1 Base_shape

AIM element: product_definition_shape
Source: ISO 10303-41
Reference Path: {product_definition_shape <=
property_definition
property_definition.definition ->
characterized_definition <-
material_designation.definitions[i]}

5.1.13.2 Block_base_shape

AIM element: block_shape_representation
Source: ISO 10303-224
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: block_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.13.2.1 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}

5.1.13.2.2 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}

5.1.13.3 B-rep_model

AIM element: manifold_solid_brep
Source: ISO 10303-42

5.1.13.4 B-rep_model_element

AIM element: (geometric_representation_item)
(topological_representation_item)
Source: ISO 10303-42

5.1.13.4.1 B-rep_model_element to B-rep_model (as element)

AIM element: PATH
Reference Path: (geometric_representation_item =>
surface <-
face_surface.face_geometry
face_surface <=
face <-)
(topological_representation_item =>
face <-)
connected_face_set.cfs_faces[i]
connected_face_set =>
closed_shell <-
manifold_solid_brep.outer
manifold_solid_brep

5.1.13.5 B-rep_shape_aspect_representation

AIM element: advanced_brep_shape_representation
Source: ISO 10303-514
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: advanced_brep_shape_representation <=
shape_representation

5.1.13.5.1 B-rep_shape_aspect_representation to B-rep_model (as shape_-definition)

AIM element: PATH
Reference Path: advanced_brep_shape_representation <=
shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
solid_model =>
manifold_solid_brep

5.1.13.6 B-rep_shape_representation

AIM element: advanced_brep_shape_representation
Source: ISO 10303-514
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: advanced_brep_shape_representation <=
shape_representation

5.1.13.6.1 B-rep_shape_representation to B-rep_model (as shape_definition)

AIM element: PATH
Reference Path: advanced_brep_shape_representation <=
shape_representation <=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
solid_model =>
manifold_solid_brep

5.1.13.7 Cylindrical_base_shape

AIM element: cylindrical_shape_representation
Source: ISO 10303-224
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: cylindrical_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.13.7.1 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}

5.1.13.8 Direction_element

AIM element: direction_shape_representation
Source: ISO 10303-522
Rules:
Reference Path: direction_shape_representation <=
shape_representation<=
representation
{representation.items[i]->
representation_item=>
geometric_representation_item=>
direction}

5.1.13.9 Explicit_base_shape_representation

AIM element: shape_representation
Source: ISO 10303-41
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)

5.1.13.9.1 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 representation'}
{representation_item
representation_item.description = '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')}

5.1.13.9.2 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 = 'explicit representation'}
{representation_item
representation_item.description = 'user defined description'}
representation_item =>
descriptive_representation_item
descriptive_representation_item.description

5.1.13.9.3 explicit_base_shape_representation to B-rep_shape_representation (as B-rep_form)

AIM element: IDENTICAL MAPPING

5.1.13.10 Face_shape_element

AIM element: face_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)5.2.4.29
Reference Path: face_shape_representation <=
shape_representation<=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
(face_surface)
(oriented_face)

5.1.13.11 Face_shape_element_relationship

AIM element: face_shape_representation_relationship
Source: ISO 10303-224
Reference Path: face_shape_representation_relationship<=
representation_relationship

5.1.13.11.1 face_shape_element_relationship to face_shape_element (as predecessor)

AIM element: PATH
Reference Path: face_shape_representation_relationship<=
representation_relationship
representation_relationship.rep_1->
representation=>
shape_representation=>
face_shape_representation

5.1.13.11.2 face_shape_element_relationship to face_shape_element (as successor)

AIM element: PATH
Reference Path: face_shape_representation_relationship<=
representation_relationship
representation_relationship.rep_2->
representation=>
shape_representation=>
face_shape_representation

5.1.13.12 Implicit_base_shape_representation

AIM element: shape_representation_with_parameters
Source: ISO 10303-522
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_representation_with_parameters <=
shape_representation

5.1.13.12.1 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}

5.1.13.12.2 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

5.1.13.13 Location_element

AIM element: location_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)5.2.4.29
Reference Path: location_shape_representation <=
shape_representation<=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'orientation'}
{representation_item =>
geometric_representation_item =>
point=>
cartesian_point}

5.1.13.14 Ngon_base_shape

AIM element: ngon_shape_representation
Source: ISO 10303-224
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: ngon_shape_representation <=
shape_representation_with_parameters <=
shape_representation

5.1.13.14.1 circumscribed_or_across_flats

AIM element: (representation_item.name = 'circumscribed diameter')
(representation_item.name =
'diameter across flats')
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')

5.1.13.14.2 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}

5.1.13.14.3 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}

5.1.13.14.4 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}

5.1.13.15 Orientation

AIM element: placement
Source: ISO 10303-42

5.1.13.15.1 axis

AIM element: (axis1_placement.axis)

((axis2_placement_3d.axis)
(axis2_placement_3d.ref_direction))
Source: ISO 10303-42
Reference Path: placement =>
(axis1_placement
axis1_placement.axis)
(axis2_placement_3d
(axis2_placement_3d.axis)
(axis2_placement_3d.ref_direction))

5.1.13.15.2 location

AIM element: placement.location
Source: ISO 10303-42

5.1.13.16 Part_placement

AIM element: mapped_item
Source: ISO 10303-43

5.1.13.16.1 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

5.1.13.16.2 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

5.1.13.16.3 part_placement to 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->
property_definition=>
product_definition_shape

5.1.13.17 Path_element

AIM element: path_shape_representation
Source: ISO 10303-522
Rules: representation_subtype_exclusiveness – (See 5.2.4.29)
Reference Path: path_shape_representation <=
shape_representation<=
representation
representation.items[i] ->
{representation_item
representation_item.name = 'profile shape'}
representation_item =>
(geometric_representation_item =>
curve=>
bounded_curve)
(topological_representation_item=>
edge=>
edge_curve)

5.1.13.18 Planar_element

AIM element: planar_shape_representation
Source: ISO 10303-522
Reference Path: planar_shape_representation <=
shape_representation<=
representation

5.1.13.18.1 location

AIM element: cartesian_point
Source: ISO 10303-42
Reference Path: planar_shape_representation<=
shape_representation<=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
surface=>
elementary_surface
{elementary_surface=>
plane}
elementary_surface.position->
axis2_placement_3d<=
placement
placement.location->
cartesian_point

5.1.13.18.2 normal

AIM element: direction
Source: ISO 10303-42
Reference Path: planar_shape_representation<=
shape_representation<=
representation
representation.items[i] ->
representation_item =>
geometric_representation_item =>
surface=>
elementary_surface
{elementary_surface=>
plane}
elementary_surface.position->
axis2_placement_3d.axis->
direction

5.1.13.19 Shape

AIM element: product_definition_shape
Source: ISO 10303-41

5.1.13.19.1 shape to base_shape (as base_shape_definition)

AIM element: PATH
Rules: material_is_specified_for_part – (See 5.2.4.18)
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 <-
material_designation.definitions[i]
material_designation}
characterized_definition <-
property_definition.definition
property_definition =>
product_definition_shape

5.1.13.19.2 shape to B-rep_shape_representation (as B-rep_form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: product_definition_shape <=
property_definition <-
property_definition_representation.definition
{property_definition_representation =>
shape_definition_representation}
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation =>
advanced_brep_shape_representation

5.1.13.19.3 shape to shape_aspect (as element)

AIM element: PATH
Reference Path: product_definition_shape <-
shape_aspect.of_shape
shape_aspect

5.1.13.20 Shape_aspect

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34

5.1.13.20.1 shape_aspect to B-rep_model_element (as B-rep_shape)

AIM element: PATH
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation
representation.items[i] ->
representation_item =>
(geometric_representation_item)
(topological_representation_item)

5.1.13.20.2 shape_aspect to B-rep_shape_aspect_representation (as B-rep_-form)

AIM element: PATH
Rules: dependent_instantiable_shape_representation – (See 5.2.4.14)
representation_subtype_exclusiveness – (See 5.2.4.29)
shape_representation_subtype_exclusiveness – (See 5.2.4.35)
Reference Path: shape_aspect
shape_definition = shape_aspect
shape_definition
characterized_definition = shape_definition
characterized_definition <-
property_definition.definition
property_definition <-
property_definition_representation.definition
property_definition_representation
property_definition_representation.used_representation ->
representation =>
shape_representation =>
advanced_brep_shape_representation

5.1.13.20.3 shape_aspect to shape_element (as element)

AIM element: IDENTICAL MAPPING

5.1.13.21 Shape_element

AIM element: shape_aspect
Source: ISO 10303-41
Rules: shape_aspect_subtype_exclusiveness – (See 5.2.4.34)5.2.4.34
Reference Path: {shape_aspect
shape_aspect.product_definitional = TRUE}

The following rules are referenced in the preceding clauses:

1) approval_requires_approval_date_time 5.2.4.2
2) approval_requires_approval_person_organization 5.2.4.3
3) dependent_instantiable_action_request_status 5.2.4.6
4) dependent_instantiable_approval_status 5.2.4.7
5) dependent_instantiable_date5.2.4.8
6) dependent_instantiable_named_unit 5.2.4.9
7) dependent_instantiable_security_classification_level 5.2.4.13
8) dependent_instantiable_shape_representation 5.2.4.14
9) geometric_tolerance_subtype_exclusiveness 5.2.4.16
10) machining_feature_life_cycle 5.2.4.17
11) material_is_specified_for_part 5.2.4.18
12) mating_definition_requires_externally_defined_representation 5.2.4.19
13) part_requires_project_order 5.2.4.23
14) part_to_approval 5.2.4.24
15) product_requires_version 5.2.4.25
16) product_definition_formation_requires_security_classification 5.2.4.26
17) project_order_requires_approval 5.2.4.27
18) project_order_tracking_relationships 5.2.4.28
19) representation_subtype_exclusiveness 5.2.4.29
20) restrict_approval_status 5.2.4.30
21) restrict_security_classification_level 5.2.4.32
22) shape_aspect_relationship_subtype_exclusiveness 5.2.4.33
23) shape_aspect_subtype_exclusiveness 5.2.4.34
24) shape_representation_subtype_exclusiveness 5.2.4.35
25) subtype_mandatory_characterized_object 5.2.4.36
26) transition_feature_life_cycle 5.2.4.37
27) transition_feature_on_part_boundary 5.2.4.38

5.2 AIM EXPRESS short listing

This clause specifies the EXPRESS schema that uses elements from the integrated resources and the AICs and contains the types, entity specializations, rules, and functions that are specific to this part of ISO 10303.This clause also specifies modifications to the textual material for constructs that are imported from the integrated resources and the AICs.The definitions and EXPRESS provided in the integrated resources for constructs used in the AIM may include select list items and subtypes which are not imported into the AIM.Requirements stated in the integrated resources which refer to such items and subtypes apply exclusively to those items which are imported into the AIM.

```
*)
SCHEMA feature_based_process_planning;

USE FROM action_schema-- ISO 10303-41
(action_directive,
 action_method,
 action_relationship,
 action_request_solution,
 action_request_status,
 action_status,
 directed_action,
```



```
 versioned_action_request);
USE FROM aic_machining_feature;                               -- ISO 10303-522
USE FROM aic_advanced_brep                                    -- ISO 10303-514
 (advanced_brep_shape_representation);

USE FROM aic_topologically_bounded_surface                    -- ISO 10303-511
 (advanced_face);

USE FROM aic_geometric_tolerances                             -- ISO 10303-519
(angularity_tolerance,
 concentricity_tolerance,
 circular_runout_tolerance,
 cylindricity_tolerance,
 flatness_tolerance,
 line_profile_tolerance,
 parallelism_tolerance,
 perpendicularity_tolerance,
 position_tolerance,
 roundness_tolerance,
 straightness_tolerance,
 surface_profile_tolerance,
 symmetry_tolerance,
 total_runout_tolerance);

USE FROM application_context_schema                    -- ISO 10303-41
(application_context,
 application_protocol_definition);

USE FROM approval_schema                               -- ISO 10303-41
(approval,
 approval_date_time,
 approval_person_organization,
 approval_status);

REFERENCE FROM basic_attribute_schema                  -- ISO 10303-41
(description_attribute,
object_role,
role_association);

USE FROM date_time_schema                              -- ISO 10303-41
(date,
 calendar_date,
 ordinal_date,
 week_of_year_and_day_date);

USE FROM document_schema                               -- ISO 10303-41
(document,
 document_representation_type,
 document_with_class,
 document_usage_constraint,
 document_relationship);

USE FROM external_reference_schema                     -- ISO 10303-41
(externally_defined_item,
 external_source,
 externally_defined_item_relationship,
 pre_defined_item);
```

```
USE FROM geometric_model_schema                       -- ISO 10303-42
(brep_with_voids,
 manifold_solid_brep);

REFERENCE FROM geometry_schema                         -- ISO 10303-42
(dummy_gri);

USE FROM geometry_schema                               -- ISO 10303-42
(axis2_placement_2d,
 axis2_placement_3d,
 bezier_curve,
 bezier_surface,
 b_spline_curve_with_knots,
 b_spline_surface_with_knots,
 cartesian_point,
 cartesian_transformation_operator_3d,
 circle,
 conical_surface,
 cylindrical_surface,
 degenerate_toroidal_surface,
 direction,
 ellipse,
 geometric_representation_context,
 hyperbola,
 line,
 parabola,
 pcurve,
 plane,
 point,
 polyline,
 quasi_uniform_curve,
 quasi_uniform_surface,
 rational_b_spline_curve,
 rational_b_spline_surface,
 spherical_surface,
 surface_curve,
 surface_of_linear_extrusion,
 surface_of_revolution,
 swept_surface,
 toroidal_surface,
 uniform_curve,
 uniform_surface,
 vector);

USE FROM group_schema                                  -- ISO 10303-41
(group,
 group_relationship);

USE FROM management_resources_schema                   -- ISO 10303-41
(action_assignment,
 action_request_assignment,
 approval_assignment,
 classification_assignment,
 date_assignment,
 document_reference,
 document_usage_constraint_assignment,
```



```
 external_identification_assignment,
 identification_assignment,
 group_assignment,
 organization_assignment,
 person_and_organization_assignment,
 security_classification_assignment);

USE FROM material_property_representation_schema      -- ISO 10303-45
(material_property_representation);

USE FROM material_property_definition_schema          -- ISO 10303-45
(material_designation,
 material_property,
 property_definition_relationship);

USE FROM measure_schema                               -- ISO 10303-41
(context_dependent_measure,
 context_dependent_unit,
 conversion_based_unit,
 count_measure,
 derived_unit,
 descriptive_measure,
 global_unit_assigned_context,
 length_measure_with_unit,
 length_unit,
 measure_with_unit,
 named_unit,
 parameter_value,
 plane_angle_measure_with_unit,
 plane_angle_unit,
 positive_plane_angle_measure,
 ratio_measure,
 ratio_measure_with_unit,
 ratio_unit,
 si_unit,
 solid_angle_measure_with_unit,
 solid_angle_unit);

USE FROM person_organization_schema                   -- ISO 10303-41
(organizational_address,
 person_and_organization,
 personal_address);

USE FROM product_definition_schema                     -- ISO 10303-41
(product,
 product_definition,
 product_definition_formation,
 product_definition_with_associated_documents);

USE FROM product_property_definition_schema           -- ISO 10303-41
(characterized_object,
 general_property,
 general_property_association,
 product_definition_shape,
 property_definition,
 shape_aspect,
 shape_aspect_relationship);
```

```
USE FROM product_property_representation_schema       -- ISO 10303-41
(property_definition_representation,
 shape_definition_representation,
 shape_representation);

USE FROM product_structure_schema                      -- ISO 10303-44
(assembly_component_usage,
 next_assembly_usage_occurrence,
 make_from_usage_option);

USE FROM qualified_measure_schema                      -- ISO 10303-45
(descriptive_representation_item,
 expanded_uncertainty,
 measure_representation_item,
 measure_qualification,
 precision_qualifier,
 qualified_representation_item,
 qualitative_uncertainty,
 standard_uncertainty,
 type_qualifier,
 uncertainty_qualifier);

USE FROM representation_schema                         -- ISO 10303-43
(definitional_representation,
 global_uncertainty_assigned_context,
 mapped_item,
 parametric_representation_context,
 compound_representation_item,
 compound_item_definition,
 list_representation_item,
 set_representation_item,
 representation,
 representation_item,
 representation_relationship,
 value_representation_item);

USE FROM security_classification_schema                -- ISO 10303-41
(security_classification,
 security_classification_level);

USE FROM shape_aspect_definition_schema                -- ISO 10303-47
(composite_shape_aspect,
 datum_feature,
 datum_reference,
 datum_target,
 referenced_modified_datum,
 datum);

USE FROM shape_dimension_schema                       -- ISO 10303-47
(angular_location,
 angular_size,
 dimensional_characteristic_representation,
 dimensional_location,
 dimensional_location_with_path,
 dimensional_size,
```



```
 dimensional_size_with_path,
 shape_dimension_representation);

USE FROM shape_tolerance_schema                          -- ISO 10303-47
(geometric_tolerance,
 geometric_tolerance_relationship,
 geometric_tolerance_with_datum_reference,
 geometric_tolerance_with_defined_unit,
 limits_and_fits,
 modified_geometric_tolerance,
 plus_minus_tolerance,
 projected_zone_definition,
 runout_zone_definition,
 runout_zone_orientation,
 tolerance_value,
 tolerance_zone,
 tolerance_zone_definition);

USE FROM support_resource_schema                        -- ISO 10303-41
(identifier,
 label,
 text);

REFERENCE FROM topology_schema                          -- ISO 10303-42
(dummy_tri);

USE FROM topology_schema                                -- ISO 10303-42
(closed_shell,
 connected_face_set,
 oriented_closed_shell,
 edge_curve,
 edge_loop,
 face_bound,
 face_outer_bound,
 face_surface,
 path,
 vertex_loop,
 vertex_point);
(*
```

NOTE The schemas referenced above can be found in the following parts of ISO 10303:

action_schema	ISO 10303-41
application_context_schema	ISO 10303-41
approval_schema	ISO 10303-41
aic_machining_feature	ISO 10303-522
aic_advanced_brep	ISO 10303-514
aic_geometric_tolerances	ISO 10303-519
aic_topologically_bounded_surface	ISO 10303-511
basic_attribute_schema	ISO 10303-41
date_time_schema	ISO 10303-41
document_schema	ISO 10303-41
external_reference_schema	ISO 10303-41
geometric_model_schema	ISO 10303-42
geometry_schema	ISO 10303-42
management_resources_schema	ISO 10303-41
material_property_definition_schema	ISO 10303-45

measure_schema	ISO 10303-41
person_organization_schema	ISO 10303-41
product_definition_schema	ISO 10303-41
product_property_definition_schema	ISO 10303-41
product_property_representation_schema	ISO 10303-41
product_structure_schema	ISO 10303-44
qualified_measure_schema	ISO 10303-45
representation_schema	ISO 10303-43
security_classification_schema	ISO 10303-41
shape_aspect_definition_schema	ISO 10303-47
shape_dimension_schema	ISO 10303-47
shape_tolerance_schema	ISO 10303-47
support_resource_schema	ISO 10303-41
topology_schema	ISO 10303-42

5.2.1 Fundamental concepts and assumptions

ISO 10303-224 is designed to be used in the representation and exchange of data to a process planning function for the manufacture of mechanical parts. This schema embodies several fundamental concepts that are manifested in the information model. The process planning function in an organization can be assisted a great deal by identifying machining oriented part shape features so that the process planner can more readily identify machining equipment, tools, and processes to manufacture a part. The fundamental concept of this schema is to enable the representation and exchange of this information about a part.

5.2.1.1 Hybrid approach to feature representation

Two ways to represent the shape of part features are specified in ISO 10303-224, explicit shape representation and implicit shape representation. The explicit representation of shape is accomplished by specifying a boundary representation solid model that defines the shape of the part feature. The implicit representation of shape is accomplished by specifying the shape defining parameters that are specified in the model for each type of part feature. The implicit representation is always dependent upon a base shape representation to which the features are applied. The schema provides the capability to use a hybrid approach to represent the shape of a part feature. This capability allows for the specification of an explicit representation, an implicit representation, or both an explicit and implicit representation for any of the part features defined for a part. The base shape representation may be specified either implicitly or explicitly as well. The explicit representation may be used in conjunction with the implicit representation to verify the interpretation of the implicit representation or to share data with another application that does not rely on the implicit representation of part features.

5.2.1.2 Use of the integrated resources for feature definition

5.2.1.2.1 Fundamental concepts and assumptions

ISO 10303-224 is designed to be used in the representation and exchange of data to a process planning function for the manufacture of mechanical parts. This schema embodies several fundamental concepts that are manifested in the information model. The process planning function in an organization can be assisted a great deal by identifying machining oriented part shape features so that the process planner can more readily identify machining equipment, tools, and processes to manufacture a part. The fundamental concept of this schema is to enable the representation and exchange of this information about a part.

5.2.1.2.2 Hybrid approach to feature representation

Two ways to represent the shape of part features are specified in ISO 10303-224, explicit shape representation and implicit shape representation. The explicit representation of shape is accomplished by specifying a boundary representation solid model that defines the shape of the part feature. The implicit representation of shape is accomplished by specifying the shape defining parameters that are specified in the model for each type of part feature. The implicit representation is always dependent upon a base shape representation to which the features are applied. The schema provides the capability to use a hybrid approach to represent the shape of a part feature. This capability allows for the specification of an explicit representation, an implicit representation, or both an explicit and implicit representation for any of the part features defined for a part. The base shape representation may be specified either implicitly or explicitly as well. The explicit representation may be used in conjunction with the implicit representation to verify the interpretation of the implicit representation or to share data with another application that does not rely on the implicit representation of part features.

5.2.1.2.3 Use of the integrated resources for feature definition

A part's machining features are elements or aspects of the shape of a part. As such, the **shape_aspect** entity from ISO 10303-41 is specialized in this schema to specify the concepts of the different types of features. Three basic types of features are specified as **instanced_feature**, **replicate_feature**, and **transition_feature**. These basic types are again specialized into the individual types of features. Two types of parameters are used to specify the implicit shape representation of a feature, a numeric and a descriptive parameter. The **measure_representation_item** is used to specify the numeric parameter providing for both a real value of length and angle or an integer value of count and the units by which each value is specified. The **descriptive_representation_item** is used to specify the descriptive parameter. The element of the implicit shape that is being defined for each feature is specified by the name of the parameter. Constraints are defined in each specialization to specify the required parameters and the value types for the implicit representation of each feature. The **shape_representation_with_-parameters** entity is used to specify the implicit representation of the shape of the feature.

Additionally, there are components of the part's features defined that are also elements of the shape of the part. As such, they are also specified using specializations of the **shape_aspect** entity. Unlike the feature representations, the feature components shall always have an implicit representation specified.

EXAMPLE 1 A profile to be swept or a taper definition are types of feature components.

The feature components define parts of the individual features and are required to be used in certain feature definitions. The feature components are related to the individual part's features through a specialization of the **shape_aspect_relationship** called the **feature_component_relationship**. Every **feature_component_relationship** specifies a component that has been defined for a particular feature.

There is also a specialization of the **shape_aspect_relationship** defined in the schema called **shape_-defining_relationship**. This entity is used to specify other elements of the shape of the part (**shape_-aspect**s) that participate in the implicit representation of the shape of the feature. These elements all have a geometric or topological representation that consists of a single **representation_item**.

EXAMPLE 2 A location or a path are types of shape elements that are shape defining.

5.2.2 Feature based process planning types

5.2.2.1 classification_item

A **classification_item** identifies an **externally_defined_representation_with_parameters**, which a referenced **classification** may be assigned.

EXPRESS specification:

```
*)
TYPE classification_item = SELECT (
     externally_defined_representation_with_parameters);
END_TYPE;

(*
```

5.2.2.2 document_reference_item

A **document_reference_item** identifies an **action_method**, **externally_defined_feature_definition**, **directed_action**, **dimensional_characteristic_representation**, **externally_defined_dimension_-definition** or **property_definition** to which a referenced document may be assigned.

EXPRESS specification:

```
*)
TYPE document_reference_item = SELECT
  (action_method,
   externally_defined_feature_definition,
   directed_action,
   dimensional_characteristic_representation,
   product_definition_formation,
   externally_defined_dimension_definition,
   property_definition);
END_TYPE;
(*
```

5.2.2.3 external_identification_item

A **external_identification_item** identifies an **document** , **externally_defined_item** or **externally_-defined_general_property** to which a referenced **external_identification** may be assigned.

EXPRESS specification:

```
*)
TYPE external_identification_item = SELECT (
     externally_defined_item,
     externally_defined_general_property);
END_TYPE;
(*
```


5.2.2.4 feature_based_pp_action_item

A **feature_based_pp_action_item** identifies a **product_definition_formation** that is being tracked by an **action_directive** for a project.

EXPRESS specification:

```
*)
TYPE feature_based_pp_action_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.5 feature_based_pp_action_request_item

A **feature_based_pp_action_request_item** identifies a **product_definition_formation** for which a design exception has been identified.

EXPRESS specification:

```
*)
TYPE feature_based_pp_action_request_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.6 feature_based_pp_approved_item

A **feature_based_pp_approved_item** identifies a **directed_action** or **product_definition_formation** to which an **approval** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_approved_item = SELECT
  (directed_action,
   product_definition_formation);
END_TYPE;
(*
```

5.2.2.7 feature_based_pp_classified_item

A **feature_based_pp_classified_item** identifies a **product_definition_formation** to which a **security_-classification** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_classified_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.8 feature_based_pp_dated_item

A **feature_based_pp_dated_item** identifies a **directed_action**, **document**, or **versioned_action_-request** to which a **date** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_dated_item = SELECT
  (directed_action,
   document,
   versioned_action_request);
END_TYPE;
(*
```

5.2.2.9 feature_based_pp_ordered_item

A **feature_based_pp_ordered_item** identifies the **product_definition_formation** that is being ordered by a customer.

EXPRESS specification:

```
*)
TYPE feature_based_pp_ordered_item = SELECT
  (product_definition_formation);
END_TYPE;
(*
```

5.2.2.10 feature_based_pp_organization_item

A **feature_based_pp_organization_item** identifies an **action_directive,** externally_defined_item, known_source or **product_definition_formation** to which an **organization** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_organization_item = SELECT
  (action_directive,
   product_definition_formation,
   externally_defined_item,
   known_source);
END_TYPE;
(*
```


5.2.2.11 feature_based_pp_person_and_organization_item

A **feature_based_pp_person_and_organization_item** identifies an **action_directive** or **product_-definition_formation** to which a **person_and_organization** may be assigned.

EXPRESS specification:

```
*)
TYPE feature_based_pp_person_and_organization_item = SELECT
  (action_directive, product_definition_formation);
END_TYPE;
(*
```

5.2.2.12 group_item

A **group_item** identifies an **instanced_feature**, **replicate_feature**, or **transition_feature** to which a referenced group may be assigned.

EXPRESS specification:

```
*)
  TYPE group_item = SELECT
    (instanced_feature, replicate_feature, transition_feature);
  END_TYPE;
(*
```

5.2.2.13 identification_item

A **identification_item** identifies an **document** to which an identification may be assigned.

EXPRESS specification:

```
*)
TYPE identification_item = SELECT (
     document);
END_TYPE;
(*
```


5.2.3 Feature based process planning entities

5.2.3.1 Feature based process planning entity definitions

5.2.3.1.1 applied_identification_assignment

A **applied_identification_assignment** specifies those **identification_item**s to which an **identification_-role** is assigned.

EXPRESS specification:

```
*)
ENTITY applied_identification_assignment
     SUBTYPE OF (identification_assignment);
     items: SET[1:?] OF identification_item;
END_ENTITY;
(*
```

5.2.3.1.2 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;
WHERE

   wr1: (NOT ( ('FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_CLASS'
        IN TYPEOF(SELF.assigned_class)) AND
        ((SELF.role.name = 'definitional class membership'))AND
        ( (SELF.assigned_class.name= 'library identifier'))));

 wr2: (SIZEOF(QUERY ( edir <* USEDIN(SELF.assigned_class,
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.'+
      'RELATED_ITEM') |
      ((edir.name='name scope')  AND
      ('FEATURE_BASED_PROCESS_PLANNING.'+
      'EXTERNALLY_DEFINED_GENERAL_PROPERTY'
      IN TYPEOF (edir.relating_item))AND
      (NOT (SIZEOF(QUERY ( gpa <* USEDIN(edir.relating_item,
      'FEATURE_BASED_PROCESS_PLANNING.'+
      'GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION')  |
      ((gpa.name='definitional') AND
      (NOT (SIZEOF (QUERY (pdr <* USEDIN(gpa.derived_definition,
      'FEATURE_BASED_PROCESS_PLANNING.'+
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ((pdr.used_representation.name='property value') AND
      ('FEATURE_BASED_PROCESS_PLANNING.VALUE_REPRESENTATION_ITEM'
      IN TYPEOF (pdr.used_representation.items))
```



```
      ))) >= 1 )) )
      )) =0))
      ))) =0);

END_ENTITY;
(*
```

Attribute definition:

items: the set of **classification_item** s to which an **classification_role** is assigned.

Formal propositions:

WR1: The **role** attribute shall define a **classificatioin_role** with a **name** of 'definitional class membership', and the **assigned_class** attribute shall define a **group** of type **externally_defined_class** with a **name** of 'library identifier'

WR2: 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 an **general_property_association** with a **name** of 'definitional' that is the **relating_item** for a **externally_defined_item_relationship** with a **name** of 'name scope'

5.2.3.1.3 applied_document_reference

A **applied_document_reference** specifies those **document_reference_item**s to which a referenced 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 a referenced document is assigned.

5.2.3.1.4 applied_document_usage_constraint_assignment

A **applied_document_usage_constraint_assignment** specifies those **document_reference_item**s to which a referenced document is assigned.

EXPRESS specification:

```
*)
ENTITY applied_document_usage_constraint_assignment
  SUBTYPE OF (document_usage_constraint_assignment);
  items : SET [1:?] OF document_reference_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **document_reference_item**s to which a referenced document is assigned.

5.2.3.1.5 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.

5.2.3.1.6 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 for which a particular **group** is applicable.

5.2.3.1.7 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. See ARM definition for Block_base_shape in paragraph 4.2.10 for more information.

EXPRESS specification:

```
*)
  ENTITY block_shape_representation
    SUBTYPE OF (shape_representation_with_parameters);
    WHERE
      wr1: (SIZEOF(SELF.items) = 4);

      wr2: (SIZEOF(QUERY ( it <* SELF.items |
           (('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT'
           IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1);

      wr3: (SIZEOF(QUERY ( it <* SELF.items |
           ((SIZEOF(
           ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);

      wr4: (SIZEOF(QUERY ( it <* SELF.items |
           ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
           TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1);

      wr5: (SIZEOF(QUERY ( it <* SELF.items |
           ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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.

5.2.3.1.8 class

A **class** is a type of **group** that specifies a type of classification assignment.

EXPRESS specification:

```
*)
ENTITY class
     SUBTYPE OF (group);
END_ENTITY;
(*
```

5.2.3.1.9 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. See ARM definition for Cylindrical_base_shape in paragraph 4.2.52 for more information.

EXPRESS specification:

```
*)
  ENTITY cylindrical_shape_representation
    SUBTYPE OF (shape_representation_with_parameters);
    WHERE
      wr1: (SIZEOF(SELF.items) = 3);

      wr2: (SIZEOF(QUERY ( it <* SELF.items | ((
           'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT'
           IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1);

      wr3: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);

      wr4: (SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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 definied by forming a circular profile in the X-Y plane, and the length along the z direction.

5.2.3.1.10 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.

5.2.3.1.11 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 marking, knurl, or thread specifications.

EXPRESS specification

```
*)
ENTITY document_file
   SUBTYPE OF (characterized_object, document);
   WHERE
 wr1: (SIZEOF(QUERY(adr<* QUERY(dr <* USEDIN(SELF,
  'DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT') |
      'DIMENSIONAL_INSPECTION_SCHEMA.APPLIED_DOCUMENT_REFERENCE'
      IN TYPEOF(dr))|
  'DIMENSIONAL_INSPECTION_SCHEMA.EXTERNALLY_DEFINED_FEATURE_DEFINITION'
      IN TYPEOF(adr.items)
      ))=1) OR
      (SIZEOF(QUERY (duc <* USEDIN(SELF,
    'DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_USAGE_CONSTRAINT.SOURCE') |
      NOT
      (SIZEOF(QUERY(aduc<* QUERY(duca <* USEDIN(duc,
 'DIMENSIONAL_INSPECTION_SCHEMA.DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.'+
      'ASSIGNED_DOCUMENT_USAGE') |
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT'
      IN TYPEOF(duca))|
  'DIMENSIONAL_INSPECTION_SCHEMA.EXTERNALLY_DEFINED_FEATURE_DEFINITION'
      IN TYPEOF(aduc.items)
      ))=1))) = 0);

 wr2: (SIZEOF(QUERY(drt <* USEDIN(SELF,
      'DIMENSIONAL_INSPECTION_SCHEMA.'+
      'DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT') |
      (drt.name='physical')))=1);
  END_ENTITY; -- document_file
(*
```

Formal propositions:

WR1: The **document_file** shall be either the **associated_document** in exactly one **applied_document_-reference** that contains one or more **external_defined_feature_definition** in its set of **items**, or a **source** in exactly one **applied_document_usage_constraint_assignment** that contains one or more **external_-defined_feature_definition** in its set of **items**.

WR2: The **document_fil**e shall be the **represented_file** in exactly one **document_representation_type** with **name** = 'physical'.

5.2.3.1.12 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,
      'FEATURE_BASED_PROCESS_PLANNING.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
      (adr.assigned_document.description =
      'externally defined size dimension specification')
      ) ) <= 1;

END_ENTITY;
(*
```

Formal propositions:

WR1: The **source** attribute shall reference a **external_reference** with a **description** of 'externally defined dimension specification''.

WR2: The **externally_defined_dimension_definition** shall be reference by a through the **items** attribute by a **document_reference** of type **applied_document_reference** that references through the **assigned_-document** attribute a **document** with **description** of 'externally defined size dimension specification'.

5.2.3.1.13 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:  'FEATURE_BASED_PROCESS_PLANNING.KNOWN_SOURCE'
  IN TYPEOF(SELF.source);
  wr2:  SELF.source.name='ISO 13584 library';

  WR3: ( SIZEOF ( QUERY ( aoa <*
       USEDIN ( SELF.source,
       'FEATURE_BASED_PROCESS_PLANNING.'+
       'FEATURE_BASED_PP_ORGANIZATION_ASSIGNMENT.ITEMS' )|
       aoa. role.name = 'library supplier' ))=1);

  WR4: ( SIZEOF ( QUERY ( aoa <*
       USEDIN ( SELF,
       'FEATURE_BASED_PROCESS_PLANNING.'+
       'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS' )|
       ('FEATURE_BASED_PROCESS_PLANNING.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 **feature_based_pp_organization_assignment** that has an **organization_role** with a **name** of 'library supplier'.

WR4: The **externally_defined_class** shall be referenced by exactly one **library_class_version_-assignment** of kind **applied_external_identification_assignment** through the **items** attribute with an **identification_role** with **name** of 'class version'.

Informal proposition:

IP1: The value of **externally_defined_class** attribute **item_id** (inherited from externally_defined_item) shall be instantiated in accordance to the class code of ISO 13584-42.

5.2.3.1.14 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:  'FEATURE_BASED_PROCESS_PLANNING.KNOWN_SOURCE'
  IN TYPEOF(SELF.source);
  wr2:  SELF.source.name='ISO 13584 library';

  WR3: ( SIZEOF ( QUERY ( aoa <*
       USEDIN ( SELF,
       'FEATURE_BASED_PROCESS_PLANNING.'+
       'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS' )|
       ('FEATURE_BASED_PROCESS_PLANNING.'+
       'LIBRARY_CLASS_VERSION_ASSIGNMENT'
       IN TYPEOF(aoa)) AND
       ( aoa. role.name = 'property version' )))=1);

  wr4: (SIZEOF(QUERY ( ap <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.'+
       'EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM') |
       ((ap.name='name scope') AND
       ('FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_CLASS'
```



```
        IN TYPEOF(ap.related_item))
        )))>=1);

END_ENTITY;
(*
```

Formal propositions:

WR1: The **source** attribute shall reference a **known_source**.

WR2: The **source** attribute shall reference a **known_source** with a **name** of 'ISO 13584 library'.

WR3: The **externally_defined_general_property** shall be referenced by exactly one **library_-class_version_assignment** of kind **applied_external_identification_assignment** through the **items** attribute with an **identification_role** with **name** of 'property version'.

WR4: The **externally_defined_general_property** shall be referenced by an **externally_defined_-item_relationship** with the **name** of 'name scope' through the relating_item that references an **externally_defined_class** through the **related_item**.

5.2.3.1.15 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

  wr1: (SIZEOF(QUERY( adr <*USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.'+
       'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') |
       'FEATURE_BASED_PROCESS_PLANNING.CLASSIFICATION_ASSIGNMENT'
       IN TYPEOF(adr))) = 1);

  wr2: (SIZEOF (QUERY(adr <* SELF.items|
        'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(adr)))=1);

  wr3: (SIZEOF (QUERY(adr <* SELF.items|
        ('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(adr)) AND
        ('FEATURE_BASED_PROCESS_PLANNING.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 exactly one **placement** in the list of **items**.

WR3: There shall be exactly one **placement** in the list of **items** with a **location** of **cartesian_point**.

5.2.3.1.16 face_shape_representation_relationship

A **face_shape_representation_relationship** is a type of **representation_relationship** that is the representation of several **face_shape_representation**s and their relationship to one another.

EXPRESS specification:

```
*)
ENTITY face_shape_representation_relationship
  SUBTYPE OF (representation_relationship);
      WHERE
      wr1: ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
            IN TYPEOF(SELF.rep_1));
      wr2: ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
            IN TYPEOF(SELF.rep_2));
END_ENTITY; -- face_shape_representation_relationship
(*
```

Formal propositions:

WR1: The **face_shape_representation_relationship** shall have only **face_shape_representation** in its **rep_1**.

WR2: The **face_shape_representation_relationship** shall have only **face_shape_representation** in its **rep_2**.

5.2.3.1.17 feature_based_pp_action_assignment

A **feature_based_pp_action_assignment** specifies those **feature_based_pp_action_item**s to which an **action** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_action_assignment
  SUBTYPE OF (action_assignment);
  items : SET [1:?] OF feature_based_pp_action_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_action_item**s for which a particular **action** is applicable.

5.2.3.1.18 feature_based_pp_action_request_assignment

A **feature_based_pp_action_request_assignment** specifies those **feature_based_pp_action_request_-items** for which a design exception has been identified.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_action_request_assignment
  SUBTYPE OF (action_request_assignment);
  items : SET [1:?] OF feature_based_pp_action_request_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_action_request_item**s for which a particular **action_request** is applicable.

5.2.3.1.19 feature_based_pp_approval_assignment

A **feature_based_pp_approval_assignment** specifies those **feature_based_pp_approved_item**s to which an **approval** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_approval_assignment
  SUBTYPE OF (approval_assignment);
  items : SET [1:?] OF feature_based_pp_approved_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_approved_item**s to which an **approval** is assigned.

5.2.3.1.20 feature_based_pp_date_assignment

A **feature_based_pp_date_assignment** specifies those **feature_based_pp_dated_item**s to which a date is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_date_assignment
  SUBTYPE OF (date_assignment);
  items : SET [1:?] OF feature_based_pp_dated_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_dated_item**s to which a date is assigned.

5.2.3.1.21 feature_based_pp_organization_assignment

A **feature_based_pp_organization_assignment** specifies those **feature_based_pp_organization_item**s to which an **organization** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_organization_assignment
  SUBTYPE OF (organization_assignment);
  items : SET [1:?] OF feature_based_pp_organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_organization_item**s to which an **organization** is assigned.

5.2.3.1.22 feature_based_pp_person_and_organization_assignment

A **feature_based_pp_person_and_organization_assignment** specifies those **feature_based_pp_-person_and_organization_item**s to which a **person_and_organization** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
  items : SET [1:?] OF feature_based_pp_person_and_organization_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_person_and_organization_item**s to which a **person_and_-organization** is assigned.

5.2.3.1.23 feature_based_pp_security_classification_assignment

A **feature_based_pp_security_classification_assignment** specifies those **feature_based_pp_-classified_item**s to which a **security_classification** is assigned.

EXPRESS specification:

```
*)
ENTITY feature_based_pp_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
```



```
  items : SET [1:?] OF feature_based_pp_classified_item;
END_ENTITY;
(*
```

Attribute definition:

items: the set of **feature_based_pp_classified_item**s to which a **security_classification** is assigned.

5.2.3.1.24 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);
  WHERE
      wr1: SELF\pre_defined_item.name = 'ISO 13584 library';
      wr2: SIZEOF(QUERY ( oa <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.'+
      'FEATURE_BASED_PP_ORGANIZATION_ASSIGNMENT.ITEMS')|
      ((NOT ('FEATURE_BASED_PROCESS_PLANNING.'+
      'ORGANIZATION_ASSIGNMENT'
      IN TYPEOF(oa))) AND (oa.role.name = 'library supplier')) ))
      = 0;
END_ENTITY;
(*
```

Formal propositions:

WR1: Each **known_source** shall have a **name** attribute with the value of 'ISO 13584 library';

WR2: Each **known_source** shall be references by at least one through the **items** attribute of the entity **feature_based_pp_organization_assignement** that references through the **role** attribute an **organization_role** with a **name** attribue of value 'library supplier'.

5.2.3.1.25 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;
(*
```

5.2.3.1.26 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;
(*
```

5.2.3.1.27 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. See ARM definition for Ngon_base_shape in paragraph 4.2.141 for more information.

EXPRESS specification:

```
*)
  ENTITY ngon_shape_representation
    SUBTYPE OF (shape_representation_with_parameters);
    WHERE
  wr1: (SIZEOF(SELF.items) = 5);

  wr2: (SIZEOF(QUERY ( it <* SELF.items |
       (('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT'
       IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1);

  wr3: (SIZEOF(QUERY ( it <* SELF.items |
       ((SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1);

  wr4: (SIZEOF(QUERY ( it <* SELF.items |
       ((SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1);

  wr5: SIZEOF (QUERY (it <* SELF.items |
       (SIZEOF
       (['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
       TYPEOF (it)) = 2) AND
     (it.name IN ['circumscribed diameter','diameter across flats']) ))= 1;

  wr6: (SIZEOF(QUERY ( it <* SELF.items |
       (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND
       ('FEATURE_BASED_PROCESS_PLANNING.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.

5.2.3.1.28 ordered_part

An **ordered_part** represents the measured amount of product that is to be manufactured.

EXPRESS specification:

```
*)
  ENTITY ordered_part
    SUBTYPE OF (action_assignment, characterized_object);
      items : SET [1:?] OF feature_based_pp_ordered_item;
    WHERE
  wr1: (SIZEOF(USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION'))= 1);

  wr2: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(USEDIN(pd,
       'FEATURE_BASED_PROCESS_PLANNING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
       = 0);

  wr3: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'FEATURE_BASED_PROCESS_PLANNING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (NOT (SIZEOF(items) = 1)) )) = 0)) )) = 0);

  wr4: (SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
       'FEATURE_BASED_PROCESS_PLANNING.' +
       'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
       (NOT (SIZEOF(QUERY ( it <* pdr.used_representation.items |
       (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
       IN TYPEOF(it)) AND (it.name = 'quantity')) )) = 1)) ))
       = 0)) )) = 0);
  END_ENTITY; -- ordered_part
(*
```

Attribute definition:

items: the set of **product_definition_formation**s for which an **action** is being assigned.

Formal propositions:

WR1: The **ordered_part** shall have exactly one property defined for it.

WR2: The property defined for the **ordered_part** shall have exactly one representation.

WR3: The representation of the property defined for the **ordered_part** shall contain exactly one **representation_item** in its set of **items**.

WR4: The **representation_item** in the set of **items** for the representation of the property for the **ordered_part** shall be of type **measure_representation_item** with a **name** of 'quantity'.

5.2.3.1.29 placed_datum_target_feature

A **placed_datum_target_feature** is a type of **datum_target** that represents the implicit definition. See ARM definition for Placed_target in paragraph 4.2.169 for more information.

EXPRESS specification:

```
*)
  ENTITY placed_datum_target_feature
    SUBTYPE OF (datum_target);
    WHERE
  wr1 : (SELF.description IN ['point','line','rectangle','circle']);
```



```
wr2 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);

wr3 : (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      (('FEATURE_BASED_PROCESS_PLANNING.PLACEMENT'
      IN TYPEOF(it)) AND (it.name = 'orientation')) )) = 1)) ))
      = 0)) )) = 0);

wr4 : ((NOT (SELF.description = 'point')) OR
      (SIZEOF(QUERY ( pd <*  USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <*
      QUERY ( pdr <* USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 1)) ))
      = 0)) )) = 0));

wr5 : ((NOT (SELF.description IN ['line','circle'])) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 2)) ))
       = 0)) )) = 0));

wr6 : ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(dtm_rep.used_representation.items) = 3)) ))
```

```
      = 0)) )) = 0));

wr7 : ((NOT (SELF.description = 'circle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target diameter')) ))
      = 1)) )) = 0)) )) = 0));

wr8 : ((NOT (SELF.description = 'line')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
      = 0)) )) = 0));

wr9 : ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      (('FEATURE_BASED_PROCESS_PLANNING.' +
      'SHAPE_REPRESENTATION_WITH_PARAMETERS')
      IN TYPEOF(pdr.used_representation)) ) |
      (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
      ((SIZEOF([
      'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
      TYPEOF(it)) = 2) AND (it.name = 'target length')))) = 1)) ))
      = 0)) )) = 0));

wr10: ((NOT (SELF.description = 'rectangle')) OR
      (SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
      (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
      'FEATURE_BASED_PROCESS_PLANNING.' +
      'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```



```
        (('FEATURE_BASED_PROCESS_PLANNING.' +
        'SHAPE_REPRESENTATION_WITH_PARAMETERS')
        IN TYPEOF(pdr.used_representation)) ) |
        (NOT (SIZEOF(QUERY ( it <* dtm_rep.used_representation.items |
        ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
        'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
        TYPEOF(it)) = 2) AND (it.name = 'target width'))))= 1))))
        = 0)) ))=0));
  END_ENTITY; -- placed_datum_target_feature
(*
```

Formal propositions:

WR1: The **description** for the **placed_datum_target_feature** shall be either 'point', 'line', 'rectangle' or 'circle'.

WR2: A **placed_datum_target_feature** shall have exactly one implicit representation.

WR3: Exactly one **representation_item** used for the representation of the **placed_datum_target_-feature** shall be of type **placement** with a **name** of 'orientation'.

WR4: If the **placed_datum_target_feature** is a point, the **representation** shall contain exactly one **representation_item**s in its set of **items**.

WR5: If the **placed_datum_target_feature** is a line or circle, the **representation** shall contain exactly two **representation_item**s in its set of **items**.
WR6: If the **placed_datum_target_feature** is a rectangle, the **representation** shall contain exactly three **representation_item**s in its set of **items**.

WR7: If the **description** of the **placed_datum_target_feature** is 'circle', exactly one **representation_-item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'target diameter'.

WR8: If the **description** of the **placed_datum_target_feature** is 'line', exactly one **representation_-item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'target length'.

WR9: If the **description** of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'target length'.

WR10: If the **description** of the **placed_datum_target_feature** is 'rectangle', exactly one **representation_item** used for the implicit representation of the **placed_datum_target_feature** shall be of type **measure_representation_item** and **length_measure_with_unit** with a **name** of 'target width'.

5.2.3.1.30 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 |
       ('FEATURE_BASED_PROCESS_PLANNING.SET_REPRESENTATION_ITEM'
        IN TYPEOF (sri))) |
        ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
        IN TYPEOF (mri))
        ))=2;

wr2:  SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
      ('FEATURE_BASED_PROCESS_PLANNING.SET_REPRESENTATION_ITEM'
       IN TYPEOF (sri))) |
   (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
    IN TYPEOF (mri))
    AND  (mri.name='lower limit'))    ))=1;

wr3:  SIZEOF(QUERY ( mri <* QUERY( sri <* SELF.item_element |
      ('FEATURE_BASED_PROCESS_PLANNING.SET_REPRESENTATION_ITEM'
      IN TYPEOF (sri))) |
  (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
    IN TYPEOF (mri))
   AND    (mri.name='upper limit'))       ))=1;

wr4: SIZEOF(QUERY( i1 <* SELF.item_element |
    ('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
    IN TYPEOF (i1)) AND
    (SIZEOF ( QUERY (i2 <* SELF.item_element |
    ('FEATURE_BASED_PROCESS_PLANNING.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 Feature based process planning imported entity modifications

5.2.3.2.1 action_relationship

The base definition of the **action_relationship** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **action_relationship** entity:

— project_order_tracking_relationships (See 5.2.4.28).

5.2.3.2.2 action_request_status

The base definition of the **action_request_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 **action_request_status** entity:

— dependent_instantiable_action_request_status (See 5.2.4.1).

5.2.3.2.3 application_context

The base definition of the **application_context** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application_context** entity:

— application_context_requires_ap_definition (See 5.2.4.1).

5.2.3.2.4 application_protocol_definition

The base definition of the **application_protocol_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **application_protocol_-definition** entity:

— application_context_requires_ap_definition (See 5.2.4.1).

5.2.3.2.5 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 rules defined in this part of ISO 10303 apply to the **approval** entity:

— approval_requires_approval_date_time (See 5.2.4.2);

— approval_requires_approval_person_organization (See 5.2.4.3).

5.2.3.2.6 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 5.2.4.2).

5.2.3.2.7 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 5.2.4.3).

5.2.3.2.8 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 rules defined in this part of ISO 10303 apply to the **approval_status** entity:

— dependent_instantiable_approval_status (See 5.2.4.7);

— restrict_approval_status (See 5.2.4.30).

5.2.3.2.9 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_mandatory_characterized_object (See 5.2.4.36).

5.2.3.2.10 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 5.2.4.8).

5.2.3.2.11 description_attribute

The base definition of the **description_attribute** 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 **description_attribute** entity:

— non_instantiable_description_attribute (See 5.2.4.20).

5.2.3.2.12 dimensional_location

The base definition of the **dimensional_location** entity is given in ISO 10303-47. The following modifications apply to this part of ISO 10303.

Attribute definitions:

relating_shape aspect: the origin of the direction of measurement of the dimension.

related_shape_aspect: the target of the direction of measurement of the dimension.

5.2.3.2.13 directed_action

The base definition of the **directed_action** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **directed_action** entity:

— project_order_tracking_relationships (See 5.2.4.28).

5.2.3.2.14 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 5.2.4.16).

5.2.3.2.15 make_from_usage_option

The base definition of the **make_from_usage_option** entity is given in ISO 10303-44. 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 **make_from_usage_option** entity:

— material_is_specified_for_part (See 5.2.4.18).

5.2.3.2.16 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 5.2.4.9).

5.2.3.2.17 object_role

The base definition of the **object_role** 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 **object_role** entity:

— non_instantiable_object_role (See 5.2.4.21).

5.2.3.2.18 precision_qualifier

The base definition of the **precision_qualifier** entity is given in ISO 10303-45. 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 5.2.4.10).

5.2.3.2.19 product

The base definition of the **product** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product** entity:

— product_requires_version (See 5.2.4.25).

5.2.3.2.20 product_definition

The base definition of the **product_definition** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rule defined in this part of ISO 10303 applies to the **product_definition** entity:

— material_is_specified_for_part (See 5.2.4.18).

5.2.3.2.21 product_definition_formation

The base definition of the **product_definition_formation** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **product_definition_formation** entity:

— part_requires_project_order (See 5.2.4.23);

— part_to_approval (See 5.2.4.24);

— product_requires_version (See 5.2.4.25);

— product_version_requires_security_classification (See 5.2.4.26).

5.2.3.2.22 representation

The base definition of the **representation** entity is given in ISO 10303-43. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 apply to the **representation** entity:

— representation_subtype_exclusiveness (See 5.2.4.29).

5.2.3.2.23 role_association

The base definition of the **role_association** 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 **role_association** entity:

— non_instantiable_role_association (See 5.2.4.22).

5.2.3.2.24 security_classification_level

The base definition of the **security_classification_level** entity is given in ISO 10303-41. The following modifications apply to this part of ISO 10303.

Associated global rules:

The following global rules defined in this part of ISO 10303 applies to the **security_classification_level** entity:

— dependent_instantiable_security_classification_level (See 5.2.4.13),

— restrict_security_classification_level (see 5.2.4.32).

5.2.3.2.25 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:

— shape_aspect_subtype_exclusiveness (See 5.2.4.34).

5.2.3.2.26 shape_aspect_relationship

The base definition of the **shape_aspect_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 **shape_aspect_relationship** entity:

— shape_aspect_relationship_subtype_exclusiveness (See 5.2.4.33).

5.2.3.2.27 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 rules defined in this part of ISO 10303 apply to the **shape_representation** entity:

— dependent_instantiable_shape_representation (See 5.2.4.14);

— shape_representation_subtype_exclusiveness (See 5.2.4.35).

5.2.3.2.28 type_qualifier

The base definition of the **type_qualifier** entity is given in ISO 10303-45. 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 5.2.4.11).

5.2.3.2.29 uncertainty_qualifier

The base definition of the **uncertainty_qualifier** entity is given in ISO 10303-45. 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 5.2.4.12).

5.2.4 Feature based process planning rules

5.2.4.1 application_context_requires_ap_definition

The **application_context_requires_ap_definition** rule specifies that each instance of **application_-context** shall be referenced by exactly one **application_protocol_definition** that specifies this part of ISO 10303.

EXPRESS specification:

```
*)
RULE application_context_requires_ap_definition FOR
  (application_context, application_protocol_definition);
WHERE
  WR1: SIZEOF (QUERY (ac <* application_context |
       NOT (SIZEOF (QUERY (apd <* application_protocol_definition |
       (ac :=: apd.application)
       AND
       (apd.application_interpreted_model_schema_name =
       'feature_based_process_planning'))) = 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 'feature_based_process_planning' as its **application_interpreted_model_schema_name**.

5.2.4.2 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.

5.2.4.3 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.

5.2.4.4 chamfer_requires_faces

The **chamfer_requires_faces** rule specifies that all instances of **chamfer** require **face_shape_-representation**.

EXPRESS specification:

```
*)
RULE              chamfer_requires_faces              FOR              (chamfer,
property_definition_representation);
WHERE
 WR1:  (SIZEOF (QUERY (cf <* chamfer |
        NOT (1 = SIZEOF (QUERY (pdr <* property_definition_representation |
       ((pdr.definition.definition = cf) AND
       ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
       IN TYPEOF (pdr.used_representation)) AND
       (pdr.used_representation.name = 'chamfer face'))
            )))
        ))=0);
END_RULE;
(*
```

Argument definitions:

chamfer: the set of all instances of **chamfer** entities.

property_definition_representation: the set of all instances of property_definition_representation entities.

Formal propositions:

WR1: The **chamfer** shall have exactly one **face_shape_representation**s in the role of the chamfer face.

5.2.4.5 chamfer_offset_requires_faces

The **chamfer_offset_requires_faces** rule specifies that all instances of **chamfer_offset** require **face_shape_representation**.

EXPRESS specification:

```
*)
RULE chamfer_offset_requires_faces FOR
      (chamfer_offset, property_definition_representation);
WHERE
 WR1:  (SIZEOF (QUERY (co <* chamfer_offset |
```



```
         NOT (1 = SIZEOF (QUERY (pdr <* property_definition_representation |
        ((pdr.definition.definition = co) AND
        ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
        IN TYPEOF (pdr.used_representation)) AND
        ( ((pdr.definition.definition.description = 'first offset') AND
        (pdr.used_representation.name = 'first face shape'))
        OR
         ((pdr.definition.definition.description = 'second offset') AND
         (pdr.used_representation.name = 'second face shape'))
              ))
           )))
        ))=0);
END_RULE;
(*
```

Argument definitions:

chamfer_offset: the set of all instances of **chamfer_offset** entities.

property_definition_representation: the set of all instances of property_definition_representation entities.

Formal propositions:

WR1: If the **description** of the **chamfer_offset** is 'first offset', the **chamfer_offset** shall have at most one one **face_shape_representation** with a **name** of 'first face shape' or if the **description** of the **chamfer_offset** is 'second offset', the **chamfer_offset** shall have at most one one **face_shape_-representation** with a **name** of 'second face shape'.

5.2.4.6 dependent_instantiable_action_request_status

The **dependent_instantiable_action_request_status** rule specifies that all instances of **action_request_-status** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE             dependent_instantiable_action_request_status             FOR
(action_request_status);
WHERE
  WR1: SIZEOF (QUERY (arst <* action_request_status |
       NOT (SIZEOF (USEDIN (arst, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

action_request_status: the set of all instances of **action_request_status**.

Formal proposition:

WR1: For each instance of **action_request_status**, there shall be a reference to the **action_request_-status** instance from an attribute of another entity.

5.2.4.7 dependent_instantiable_approval_status

The **dependent_instantiable_approval_status** rule specifies that all instances of **approval_status** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
       NOT (SIZEOF (USEDIN (ast, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

approval_status: the set of all instances of **approval_status**.

Formal proposition:

WR1: For each instance of **approval_status**, there shall be a reference to the **approval_status** instance from an attribute of another entity.

5.2.4.8 dependent_instantiable_date

The **dependent_instantiable_date** rule specifies that all instances of **date** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_date FOR (date);
WHERE
  WR1: SIZEOF (QUERY (dt <* date |NOT(SIZEOF (USEDIN (dt,'')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

date: the set of all instances of **date**.

Formal proposition:

WR1: For each instance of **date**, there shall be a reference to the **date** instance from an attribute of another entity.

5.2.4.9 dependent_instantiable_named_unit

The **dependent_instantiable_named_unit** rule specifies that all instances of **named_unit** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_named_unit FOR (named_unit);
WHERE
  WR1: SIZEOF (QUERY (nu <* named_unit |
       NOT (SIZEOF (USEDIN (nu, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

named_unit: the set of all instances of **named_unit**.

Formal proposition:

WR1: For each instance of **named_unit**, there shall be a reference to the **named_unit** instance from an attribute of another entity.

5.2.4.10 dependent_instantiable_precision_qualifier

The **dependent_instantiable_precision_qualifier** rule specifies that all instances of **precision_qualifier** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_precision_qualifier FOR (precision_qualifier);
WHERE
  WR1: SIZEOF (QUERY (pq <* precision_qualifier |
       NOT (SIZEOF (USEDIN (pq, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

precision_qualifier: the set of all instances of **precision_qualifier**.

Formal proposition:

WR1: For each instance of **precision_qualifier**, there shall be a reference to the **precision_qualifier** instance from an attribute of another entity.

5.2.4.11 dependent_instantiable_type_qualifier

The **dependent_instantiable_type_qualifier** rule specifies that all instances of **type_qualifier** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_type_qualifier FOR (type_qualifier);
WHERE
  WR1: SIZEOF (QUERY (tq <* type_qualifier |
       NOT (SIZEOF (USEDIN (tq, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

type_qualifier: the set of all instances of **type_qualifier**.

Formal proposition:

WR1: For each instance of **type_qualifier**, there shall be a reference to the **type_qualifier** instance from an attribute of another entity.

5.2.4.12 dependent_instantiable_uncertainty_qualifier

The **dependent_instantiable_uncertainty_qualifier** rule specifies that all instances of **uncertainty_qualifier** are dependent on the usage to define another entity.
EXPRESS specification:

```
*)
RULE            dependent_instantiable_uncertainty_qualifier            FOR
(uncertainty_qualifier);
WHERE
  WR1: SIZEOF (QUERY (uq <* uncertainty_qualifier |
       NOT (SIZEOF (USEDIN (uq, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

uncertainty_qualifier: the set of all instances of **uncertainty_qualifier**.

Formal proposition:

WR1: For each instance of **uncertainty_qualifier**, there shall be a reference to the **uncertainty_qualifier** instance from an attribute of another entity.

5.2.4.13 dependent_instantiable_security_classification_level

The **dependent_instantiable_security_classification_level** rule specifies that all instances of **security_classification_level** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_security_classification_level FOR
  (security_classification_level);
```



```
WHERE
  WR1: SIZEOF (QUERY (scl <* security_classification_level |
       NOT (SIZEOF (USEDIN (scl, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

security_classification_level: the set of all instances of **security_classification_level**.

Formal proposition:

WR1: For each instance of **security_classification_level**, there shall be a reference to the **security_-classification_level** instance from an attribute of another entity.

5.2.4.14 dependent_instantiable_shape_representation

The **dependent_instantiable_shape_representation** rule specifies that all instances of **shape_-representation** are dependent on the usage to define another entity.

EXPRESS specification:

```
*)
RULE dependent_instantiable_shape_representation FOR (shape_representation);
WHERE
  WR1: SIZEOF (QUERY (sr <* shape_representation |
       NOT (SIZEOF (USEDIN (sr, '')) >= 1))) = 0;
END_RULE;
(*
```

Argument definition:

shape_representation: the set of all instances of **shape_representation**.

Formal proposition:

WR1: For each instance of **shape_representation**, there shall be a reference to the **shape_-representation** instance from an attribute of another entity.

5.2.4.15 edge_round_requires_faces

The **edge_round_requires_faces** rule specifies that all instances of **edge_round** require **face_shape_-representation**.

EXPRESS specification:

```
*)
RULE edge_round_requires_faces FOR
      (edge_round, property_definition_representation);
WHERE
 WR1:  (0 = SIZEOF (QUERY (er <* edge_round |
        NOT (1 = SIZEOF (QUERY (pdr <* property_definition_representation |
```



```
                ((pdr.definition.definition = er) AND
                 ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
                   IN TYPEOF (pdr.used_representation)) AND
                 (pdr.used_representation.name = 'edge round face'))
            )))
        )));

 WR2:  (SIZEOF (QUERY (er <* edge_round |
        NOT (1 = SIZEOF (QUERY (pdr <* property_definition_representation |
                ((pdr.definition.definition = er) AND
                 ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
                   IN TYPEOF (pdr.used_representation)) AND
                 (pdr.used_representation.name = 'first face shape'))
            )))
        ))=0);

 WR3:  (SIZEOF (QUERY (er <* edge_round |
        NOT (1 = SIZEOF (QUERY (pdr <* property_definition_representation |
                ((pdr.definition.definition = er) AND
                 ('FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
                   IN TYPEOF (pdr.used_representation)) AND
                 (pdr.used_representation.name = 'second face shape'))
            )))
        ))=0);
END_RULE;
(*
```

Argument definitions:

edge_round: the set of all instances of **edge_round** entities.

property_definition_representation: the set of all instances of **property_definition_representation** entities.

Formal propositions:

WR1: The **edge_round** shall have exactly one one **face_shape_representation** with a **name** of 'edge round face'.

WR2: The **edge_round** shall have exactly one one **face_shape_representation** with a **name** of 'first face shape'.

WR3: The **edge_round** shall have exactly one one **face_shape_representation** with a **name** of 'second face shape'.

5.2.4.16 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) *
       ['FEATURE_BASED_PROCESS_PLANNING.ANGULARITY_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_RUNOUT_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.CONCENTRICITY_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.CYLINDRICITY_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.FLATNESS_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.LINE_PROFILE_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.PARALLELISM_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.PERPENDICULARITY_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.POSITION_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.ROUNDNESS_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.STRAIGHTNESS_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.SURFACE_PROFILE_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.SYMMETRY_TOLERANCE',
       'FEATURE_BASED_PROCESS_PLANNING.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**.

5.2.4.17 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'.

5.2.4.18 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, 'FEATURE_BASED_PROCESS_PLANNING.' +
       '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**.

5.2.4.19 mating_definition_requires_externally_defined_representation

The **mating_definition_requires_externally_defined_representation** specifies that an instance of **product_definition** may have a reference to a **externally_defined_representation_with_parameters** to define external references to a library of part definitions.

EXPRESS specification:

```
*)
RULE mating_definition_requires_externally_defined_representation FOR
```



```
  (product_definition,
   externally_defined_representation_with_parameters);
WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition |
     ((pdf.frame_of_reference.name = 'mating definition') AND
     (NOT (SIZEOF(QUERY ( pd <* USEDIN(pdf,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
     (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
     'FEATURE_BASED_PROCESS_PLANNING.'+
     'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('FEATURE_BASED_PROCESS_PLANNING.'+
     'EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS')
     IN TYPEOF(pdr.used_representation)) ) |
     (NOT (SIZEOF(dtm_rep.used_representation.items)
     >= 0)) )) =0 )) )) = 0 )) )) ) = 0;

END_RULE;
(*
```

Formal propositions:

WR1: For each instance of **product_definition** that references a **product_definiton_formation** that references a **product** that references a **product_context** with name 'mating definition' there shall be zero , one, or many references to an externally_defined_representation_with_parameters.

5.2.4.20 non_instantiable_description_attribute

The **non_instantiable_description_attribute** rule specifies that no instances of **description_attribute** are making a reference to another entity.

EXPRESS specification:

```
*)
  RULE non_instantiable_description_attribute FOR (
           description_attribute);
  WHERE
    wr1: (SIZEOF(QUERY ( da <* description_attribute | (NOT (SIZEOF(
           USEDIN(da,'')) = 1)) )) = 0);
  END_RULE; -- non_description_attribute
(*
```

Argument definition:

description_attribute: the set of all instances of **description_attribute**.

5.2.4.21 non_instantiable_object_role

The **non_instantiable_object_role** rule specifies that no instances of **object_role** are making a reference to another entity.

EXPRESS specification:

```
*)
```



```
  RULE non_instantiable_object_role FOR (
             object_role);

  WHERE
    wr1: (SIZEOF(QUERY ( da <* object_role | (NOT (SIZEOF(
             USEDIN(da,'')) = 1)) )) = 0);
  END_RULE; --  non_object_role
(*
```

Argument definition:

object_role: the set of all instances of **object_role**.

5.2.4.22 non_instantiable_role_association

The **non_instantiable_role_association** rule specifies that no instances of **role_association** are making a reference to another entity.

EXPRESS specification:

```
*)
  RULE non_instantiable_role_association FOR (
             role_association);
  WHERE
    wr1: (SIZEOF(QUERY ( da <* role_association | (NOT (SIZEOF(
             USEDIN(da,'')) = 1)) )) = 0);
  END_RULE; --   non_role_association
(*
```

Argument definition:

role_association: the set of all instances of **role_association**.

5.2.4.23 part_requires_project_order

The **part_requires_project_order** rule specifies that each instance of **product_definition_formation** shall be referenced by exactly one instance of **feature_based_pp_action_assignment** that assigns a project order to the part.

EXPRESS specification:

```
*)
RULE part_requires_project_order FOR (product_definition_formation,
  feature_based_pp_action_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (fbppaa <* feature_based_pp_action_assignment |
       (pdf IN fbppaa.items) AND
       (fbppaa.assigned_action.name = 'project order'))) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_action_assignment: the set of all instances of **feature_based_pp_action_-assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_action_assignment** that contains the instance of **product_definition_formation** in its set of **items** and references an **action** with a name of 'project order'.

5.2.4.24 part_to_approval

The **part_to_approval** rule specifies that each instance of **product_definition_formation** shall be referenced by at most one instance of **feature_based_pp_approval_assignment**.

EXPRESS specification:
```
*)
RULE part_to_approval FOR (product_definition_formation,
  feature_based_pp_approval_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (fbppa <* feature_based_pp_approval_assignment |
       pdf IN fbppa.items )) <= 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_approval_assignment: the set of all instances of **feature_based_pp_approval_-assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_approval_assignment** that contains the instance of **product_definition_formation** in its set of **items**.

5.2.4.25 product_requires_version

The **product_requires_version** rule specifies that each instance of **product** shall be referenced by at least one instance of **product_definition_formation**. This rule enforces the requirement for every product to have one or more versions.

EXPRESS specification:

```
*)
RULE product_requires_version FOR (product, product_definition_formation);
WHERE
  WR1: SIZEOF (QUERY (prod <* product |
       NOT (SIZEOF (QUERY (pdf <* product_definition_formation |
       prod :=: pdf.of_product )) >= 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

product: the set of all instances of **product** entities.

product_definition_formation: the set of all instances of **product_definition_formation** entities.

Formal propositions:

WR1: For each instance of **product**, there shall be one or more instances of **product_definition_-formation** that contains an **of_product** attribute value equal to that instance of **product**.

5.2.4.26 product_definition_formation_requires_security_classification

The **product_definition_formation_requires_security_classification** rule specifies that each instance of **product_definition_formation** shall be referenced by exactly one instance of **feature_based_pp_-security_classification_assignment**. This rule enforces the requirement for every version of a design to have a security classification.

EXPRESS specification:

```
*)
RULE product_definition_formation_requires_security_classification FOR
  (product_definition_formation,
  feature_based_pp_security_classification_assignment);
WHERE
  WR1: SIZEOF (QUERY (pdf <* product_definition_formation |
       NOT (SIZEOF (QUERY (fbppsca <*
       feature_based_pp_security_classification_assignment |
       pdf IN fbppsca.items )) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

product_definition_formation: the set of all instances of **product_definition_formation** entities.

feature_based_pp_security_classification_assignment: the set of all instances of **feature_based_pp_-security_classification_assignment** entities.

Formal propositions:

WR1: For each instance of **product_definition_formation**, there shall be exactly one instance of **feature_based_pp_security_classification_assignment** that contains the instance of **product_-definition_formation** in its set of **items**.

5.2.4.27 project_order_requires_approval

The **project_order_requires_approval** rule specifies that each instance of **directed_action** that references an **action** with a **name** of 'project order' shall be referenced by exactly one instance of **feature_based_pp_approval_assignment**. This rule enforces the requirement for every project order to have an approval.

EXPRESS specification:

```
*)
RULE project_order_requires_approval FOR
  (directed_action, feature_based_pp_approval_assignment);
WHERE
  WR1: SIZEOF (QUERY (po <* QUERY (da <* directed_action |
       da.name = 'project order') |
      NOT (SIZEOF (QUERY (fbppapp <* feature_based_pp_approval_assignment |
       po IN fbppapp.items )) = 1 ))) = 0;
END_RULE;
(*
```

Argument definitions:

directed_action: the set of all instances of **directed_action** entities.

feature_based_pp_approval_assignment: the set of all instances of **feature_based_pp_approval_-assignment** entities.

Formal propositions:

WR1: For each instance of **directed_action** there shall be exactly one instance of **feature_based_pp_-approval_assignment** that contains the instance of **directed_action** in its set of **items**.

5.2.4.28 project_order_tracking_relationships

The **project_order_tracking_relationships** rule specifies the relationship between a project order and the other types of orders within the process planning preparation. Each order shall be carried out under a project order, and an individual project order shall be defined for tracking at most one of each type of order within the project. The types of orders defined for this part of ISO 10303 are shop work order, resource acquisition order, digital technical data package work order, and pedigree creation order.

EXPRESS specification:

```
*)
RULE project_order_tracking_relationships FOR (directed_action,
  action_relationship);
WHERE
```

```
  WR1: SIZEOF (QUERY (da <* directed_action |
       (da.name IN ['shop work order', 'resource acquisition order',
       'digital technical data package work order',
       'pedigree creation order']) AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.related_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.relating_action)) AND
       (ar.relating_action.name = 'project order')))) = 1))) = 0;
  WR2: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'project order') AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.relating_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.related_action)) AND
       (ar.related_action.name  = 'shop work order')))) <= 1))) = 0;
  WR3: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'project order') AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.relating_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.related_action)) AND
       (ar.related_action.name  = 'resource acquisition order'))))
       <= 1))) = 0;
  WR4: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'project order') AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.relating_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.related_action)) AND
       (ar.related_action.name  =
       'digital technical data package work order')))) <= 1))) = 0;
  WR5: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'project order') AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.relating_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.related_action)) AND
       (ar.related_action.name  = 'pedigree creation order'))))
       <= 1))) = 0;
  WR6: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'customer order') AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.related_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.relating_action)) AND
       (ar.relating_action.name = 'project order')))) >= 1))) = 0;
  WR7: SIZEOF (QUERY (da <* directed_action |
       (da.name = 'project order' ) AND
       NOT (SIZEOF (QUERY (ar <* action_relationship |
       (da :=: ar.relating_action) AND
       (('FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN
       TYPEOF (ar.related_action)) AND
       (ar.related_action.name  = 'customer order'))))
       <= 1))) = 0;
END_RULE;
```

```
(*
```

Argument definitions:

directed_action: the set of all instances of **directed_action** entities.

action_relationship: the set of all instances of **action_relationship** entities.

Formal propositions:

WR1: For each instance of **directed_action** with a **name** of either 'shop work order', 'resource acquisition order', 'digital technical data package work order', or 'pedigree creation order', there shall be exactly one instance of **action_relationship** that references that **directed_action** as the **related_action**, and references an instance of **directed_action** with a **name** of 'project order' as the **relating_action**.

WR2: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'shop work order'.

WR3: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'resource acquisition order'.

WR4: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'digital technical data package work order'.

WR5: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'pedigree creation order'.

WR6: For each instance of **directed_action** with a **name** of 'customer order' there shall be at least one instance of **action_relationship** that references that **directed_action** as the **related_action**, and references an instance of **directed_action** with a **name** of 'project order' as the **relating_action**.

WR7: For each instance of **directed_action** with a **name** of 'project order', there shall be at most one instance of **action_relationship** that references that **directed_action** as the **relating_action**, and references an instance of **directed_action** with a **name** of 'customer order'.

5.2.4.29 representation_subtype_exclusiveness

The **representation_subtype_exclusiveness** rule specifies that an instance of the subtypes of a **representation** shall be only one of **shape_representation**, **definitional_representation**, **direction_-shape_representation**, **face_shape_representation**, **location_shape_representation**, **planar_shape_-representation** or **path_shape_representation**.

EXPRESS specification:

```
*)
RULE representation_subtype_exclusiveness FOR (representation);
WHERE
```



```
  WR1: SIZEOF (QUERY (rep <* representation |
       NOT (SIZEOF (TYPEOF (rep) *
       ['FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.DEFINITIONAL_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.LOCATION_SHAPE_REPRESENTATION',
       'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'])
       <= 2))) = 0;
END_RULE;
(*
```

Argument definitions:

representation: the set of all instances of **representation** entities.

Formal propositions:

WR1: Each instance of the subtypes of **representation** shall be one of the **shape_representation**, **-definitional_representation**, **direction_shape_representation**, **face_shape_representation**, **location_-shape_representation**, **planar_shape_representation** or **path_shape_representation** subtypes.

5.2.4.30 restrict_approval_status

The **restrict_approval_status** rule specifies that the only values of **approval_status** permitted shall be 'approved', 'not_yet_approved', 'disapproved', or 'withdrawn'.

EXPRESS specification:

```
*)
RULE restrict_approval_status FOR (approval_status);
WHERE
  WR1: SIZEOF (QUERY (ast <* approval_status |
       NOT (ast.name IN
       ['approved', 'not yet approved', 'disapproved', 'withdrawn']))) = 0;
END_RULE;
(*
```

Argument definitions:

approval_status: the set of all instances of **approval** entities.

Formal propositions:

WR1: For each instance of **approval**, the value of the **approval_status** attribute shall be either 'approved', 'not yet approved', 'disapproved', or 'withdrawn'.

Attribute value definitions:

approved: specifies that the required authorizations have been obtained for a particular role of approval for a piece of product data.

not_yet_approved: specifies that the required authorizations are being awaited for a particular role of approval for a piece of product data.

disapproved: specifies that the required authorizations have been denied for a particular role of approval for a piece of product data.

withdrawn: specifies that the required authorizations have been revoked for a particular role of approval for a piece of product data.

5.2.4.31 restrict_externally_defined_feature_definition

The **restrict_externally_defined_feature_definition** is a global rule that restricts the use of **externally_-defined_feature_definition**.

EXPRESS specification

```
*)
RULE restrict_externally_defined_feature_definition
  FOR(EXTERNALLY_DEFINED_FEATURE_DEFINITION);
WHERE
      wr1:((SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
         (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
         'FEATURE_BASED_PROCESS_PLANNING.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
         ('FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE'
         IN TYPEOF(adr.assigned_document)))) >=1)))) =0) OR
         (SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
         (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
         'FEATURE_BASED_PROCESS_PLANNING'+
         '.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS')|
         ('FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE'
         IN TYPEOF(
  adr\document_usage_constraint_assignment.assigned_document_usage.source))))
         >=1)))) =0));END_RULE;
(*
```

Formal propositions:

WR1: The **externally_defined_feature_definition** shall be in the set of **items** of more than one **applied_document_reference** or exactly one **applied_document_usage_constraint_assignment** that defines the **document_file** containing the feature specification.

5.2.4.32 restrict_security_classification_level

The **restrict_security_classification_level** rule specifies the permitted levels of security. This rule enforces the requirement for the levels of security to be "unclassified", "classified", "proprietary", "confidential", "secret", or "top_secret".

EXPRESS specification

```
*)
RULE restrict_security_classification_level FOR
```

```
  (security_classification_level);
WHERE
  WR1: SIZEOF (QUERY (scl <* security_classification_level |
       NOT (scl.name IN
       ['unclassified', 'classified', 'proprietary','confidential', 'secret',
       'top_secret']))) = 0;
END_RULE;
(*
```

Argument definitions:

security_classification_level: identifies the set of all instances of **security_classification_level** entities.

Formal propositions:

WR1: For each instance of **security_classification_level**, the **name** attribute shall contain a value of "unclassified", "classified", "proprietary", "confidential", "secret", or "top_secret".

Attribute value definitions:

unclassified: identifies the classification level for which no security is necessary.

classified: identifies the classification level for which security is necessary, but the classification details are not given.

proprietary: identifies the classification level for which the disclosure of information about the part or the design of the part would risk an organization's market or competitive advantage.

confidential: identifies the classification level for which the disclosure of information about the part or the design of the part would cause damage to national or organizational security.

secret: identifies the classification level for which the disclosure of information about the part or the design of the part would cause serious damage to national or organizational security.

top_secret: identifies the classification level for which the disclosure of information about the part or the design of the part would cause exceptionally grave damage to national or organizational security.

5.2.4.33 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) *
       ['FEATURE_BASED_PROCESS_PLANNING.DIMENSIONAL_LOCATION',
```



```
        'FEATURE_BASED_PROCESS_PLANNING.GEOMETRIC_TOLERANCE_RELATIONSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP',
        'FEATURE_BASED_PROCESS_PLANNING.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**.

5.2.4.34 shape_aspect_subtype_exclusiveness

The **shape_aspect_subtype_exclusiveness** rule specifies that an instance of the subtypes of **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**, **instanced_feature**, **replicate_feature**, **transition_feature**, **datum**, **datum_feature**, **common_datum** or **datum_target**.
EXPRESS specification:

```
*)
RULE shape_aspect_subtype_exclusiveness FOR (shape_aspect);
  WHERE
    wr1: (SIZEOF(QUERY ( sr <* shape_aspect | (NOT (SIZEOF(TYPEOF(sr) * [
             'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT',
             'FEATURE_BASED_PROCESS_PLANNING.SLOT_END',
             'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM',
             'FEATURE_BASED_PROCESS_PLANNING.PROFILE_FLOOR',
             'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP_FLOOR',
             'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP',
             'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM',
             'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA',
             'FEATURE_BASED_PROCESS_PLANNING.TAPER',
             'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET',
             'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE',
             'FEATURE_BASED_PROCESS_PLANNING.INSTANCED_FEATURE',
             'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
```

```
              'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE',
              'FEATURE_BASED_PROCESS_PLANNING.COMMON_DATUM',
              'FEATURE_BASED_PROCESS_PLANNING.DATUM',
              'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
              'FEATURE_BASED_PROCESS_PLANNING.DATUM_TARGET']) <= 2)) ))
              = 0);
  END_RULE; -- shape_aspect_subtype_exclusiveness
(*
```

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 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**, **instanced_feature**, **replicate_feature**, **transition_feature**, **datum**, **datum_feature**, or **datum_target**.

5.2.4.35 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) *
      ['FEATURE_BASED_PROCESS_PLANNING.ADVANCED_BREP_SHAPE_REPRESENTATION',
     'FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS',
     'FEATURE_BASED_PROCESS_PLANNING.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**.

5.2.4.36 subtype_mandatory_characterized_object

The **subtype_mandatory_characterized_object** rule specifies the permitted usage of the **characterized_object**. The **characterized_object** entity shall be limited to its use in the definition of a **feature_definition**, **feature_component_definition**, or **ordered_part**.

EXPRESS specification:

```
*)

  RULE subtype_mandatory_characterized_object FOR (characterized_object);
  WHERE
    wr1: ((SIZEOF(QUERY ( csa <* characterized_object |
          (NOT (SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE',
          'FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION',
          'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION',
          'FEATURE_BASED_PROCESS_PLANNING.ORDERED_PART'] *
          TYPEOF(csa)) = 1))
          AND
          (NOT(SIZEOF(QUERY ( pd <* USEDIN(csa,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
          ('FEATURE_BASED_PROCESS_PLANNING.MATERIAL_PROPERTY'
          IN TYPEOF(pd)) )) = 1))
          )) = 0) );
  END_RULE; -- subtype_mandatory_characterized_object
(*
```

Argument definitions:

characterized_object: the set of all instances of **characterized_object** entities.
Formal propositions:

WR1: Each instance of **characterized_object** shall be either a **feature_definition**, **feature_-component_definition**, or **ordered_part**, or referenced by the **definition** attribute of **material_property**.

5.2.4.37 transition_feature_life_cycle

The **transition_feature_life_cycle** rule specifies that each instance of **transition_feature** shall be defined for the manufacturing planning stage of the part on which it is specified.

EXPRESS specification:

```
*)
RULE transition_feature_life_cycle FOR
 (transition_feature);
WHERE
  WR1: SIZEOF (QUERY (tf <* transition_feature |
       NOT (tf.of_shape.definition.frame_of_reference.life_cycle_stage =
       'manufacturing planning'))) = 0;
END_RULE;
(*
```

Argument definitions:

transition_feature: the set of all instances of **transition_feature** entities.

Formal propositions:

WR1: For each instance of **transition_feature**, the **life_cycle_stage** of the **product_definition** for which it is defined has a value of 'manufacturing planning'.

5.2.4.38 transition_feature_on_part_boundary

The **transition_feature_on_part_boundary** rule specifies that each instance of **transition_feature** shall lie on the boundary of the part for which it is defined.

EXPRESS specification:

```
*)
RULE transition_feature_on_part_boundary FOR
 (transition_feature);
WHERE
  WR1: SIZEOF (QUERY (tf <* transition_feature |
       NOT (tf.product_definitional))) = 0;
END_RULE;
(*
```

Argument definitions:

transition_feature: the set of all instances of **transition_feature** entities.

Formal propositions:

WR1: For each instance of **transition_feature**, **product_definitional** shall have a value of true.

```
*)
END_SCHEMA;                                  -- feature_based_process_planning
(*
```

6 Conformance requirements

Conformance to this part of ISO 10303 includes satisfying the requirements stated in this part, the requirements of the implementation methods 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 are specified in Annex C.

The Protocol Information 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 only one option that may be supported by an implementation. This option shall all be supported by a single class of conformance which consist of all the units of functionality for this part of ISO 10303.

This conformance class is characterized as follows:

— Feature based process planning and shape represented by advanced b-rep.

Annex A
(normative)
AIM EXPRESS expanded listing

The following EXPRESS is the expanded form of the short form schema given in 5.2. In the event of any discrepancy between the short form and this expanded listing, the expanded listing shall be used.

```
SCHEMA feature_based_process_planning;

  CONSTANT
  dummy_gri : geometric_representation_item := representation_item('') ||
                 geometric_representation_item();
  dummy_tri : topological_representation_item := representation_item('')
                 || topological_representation_item();
  END_CONSTANT;

  TYPE angle_relator = ENUMERATION OF
    (equal,
     small,
     large);
  END_TYPE; -- angle_relator

  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 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 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 context_dependent_measure = REAL;
END_TYPE; -- context_dependent_measure

TYPE count_measure = NUMBER;
END_TYPE; -- count_measure

TYPE curve_on_surface = SELECT
  (pcurve,
   surface_curve);
END_TYPE; -- curve_on_surface

TYPE date_time_or_event_occurrence = SELECT
  (date_time_select);
END_TYPE; -- date_time_or_event_occurrence

TYPE date_time_select = SELECT
  (date);
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 day_in_week_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 7);
```



```
END_TYPE; -- day_in_week_number

TYPE day_in_year_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 366);
END_TYPE; -- day_in_year_number

TYPE derived_property_select = SELECT
  (property_definition);
END_TYPE; -- derived_property_select

TYPE description_attribute_select = SELECT
  (action_request_solution,
   application_context,
   approval_role,
   date_role,
   external_source,
   organization_role,
   person_and_organization_role,
   person_and_organization,
   property_definition_representation,
   representation);
END_TYPE; -- description_attribute_select

TYPE descriptive_measure = STRING;
END_TYPE; -- descriptive_measure

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,
   externally_defined_feature_definition,
   directed_action,
   dimensional_characteristic_representation,
   product_definition_formation,
   externally_defined_dimension_definition,
   property_definition);
END_TYPE; -- document_reference_item

TYPE external_identification_item = SELECT
  (externally_defined_item,
   externally_defined_general_property);
END_TYPE; -- external_identification_item

TYPE feature_based_pp_action_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_action_item
```

```
TYPE feature_based_pp_action_request_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_action_request_item

TYPE feature_based_pp_approved_item = SELECT
  (directed_action,
   product_definition_formation);
END_TYPE; -- feature_based_pp_approved_item

TYPE feature_based_pp_classified_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_classified_item

TYPE feature_based_pp_dated_item = SELECT
  (directed_action,
   document,
   versioned_action_request);
END_TYPE; -- feature_based_pp_dated_item

TYPE feature_based_pp_ordered_item = SELECT
  (product_definition_formation);
END_TYPE; -- feature_based_pp_ordered_item

TYPE feature_based_pp_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE; -- feature_based_pp_organization_item

TYPE feature_based_pp_person_and_organization_item = SELECT
  (action_directive,
   product_definition_formation);
END_TYPE; -- feature_based_pp_person_and_organization_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
  (instanced_feature,
   replicate_feature,
   transition_feature);
END_TYPE; -- group_item

TYPE id_attribute_select = SELECT
  (action,
   address,
   property_definition,
   shape_aspect,
   shape_aspect_relationship,
   application_context,
```

```
    group,
    representation);
  END_TYPE; -- id_attribute_select

  TYPE identification_item = SELECT
    (document);
  END_TYPE; -- identification_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 measure_value = SELECT
    (length_measure,
     plane_angle_measure,
     ratio_measure,
     parameter_value,
     context_dependent_measure,
     descriptive_measure,
     positive_length_measure,
     positive_plane_angle_measure,
     count_measure);
  END_TYPE; -- measure_value

  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,
   person_and_organization,
   product_definition,
   property_definition_representation);
END_TYPE; -- name_attribute_select

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 product_or_formation_or_definition = SELECT
  (product,
   product_definition_formation,
   product_definition);
END_TYPE; -- product_or_formation_or_definition

TYPE ratio_measure = REAL;
END_TYPE; -- ratio_measure

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,
   document_reference,
   group_assignment,
   security_classification_assignment);
END_TYPE; -- role_select

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
  (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 source_item = SELECT
   (identifier);
END_TYPE; -- source_item

TYPE supported_item = SELECT
   (action_directive,
    action,
    action_method);
END_TYPE; -- supported_item

TYPE text = STRING;
END_TYPE; -- text

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_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 week_in_year_number = INTEGER;
WHERE
  wr1: (1 <= SELF) AND (SELF <= 53);
END_TYPE; -- week_in_year_number

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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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_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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,'FEATURE_BASED_PROCESS_PLANNING.' +
             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    wr2: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
             'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
END_ENTITY; -- action_request_solution

ENTITY action_request_status;
    status           : label;
    assigned_request : versioned_action_request;
END_ENTITY; -- action_request_status
```

```
ENTITY action_status;
    status          : label;
    assigned_action : executed_action;
END_ENTITY; -- action_status

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([
             'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
             'FEATURE_BASED_PROCESS_PLANNING.FACETED_BREP',
             'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.AXIS2_PLACEMENT_3D'] *
             TYPEOF(it)) = 1)) )) = 0;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP',
             'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM'] * TYPEOF(it))
             = 1) )) > 0;
    wr3: SIZEOF(QUERY ( msb <* QUERY ( it <* SELF.items | (
             'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN
             TYPEOF(it)) ) | (NOT (SIZEOF(QUERY ( csh <* msb_shells(msb)
              | (NOT (SIZEOF(QUERY ( fcs <* csh\connected_face_set.
             cfs_faces | (NOT (
         'FEATURE_BASED_PROCESS_PLANNING.ADVANCED_FACE' IN TYPEOF(fcs))) ))
             = 0)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( msb <* QUERY ( it <* items | (
             'FEATURE_BASED_PROCESS_PLANNING.MANIFOLD_SOLID_BREP' IN
             TYPEOF(it)) ) | (
             'FEATURE_BASED_PROCESS_PLANNING.ORIENTED_CLOSED_SHELL' IN
             TYPEOF(msb\manifold_solid_brep.outer)) )) = 0;
    wr5: SIZEOF(QUERY ( brv <* QUERY ( it <* items | (
```



```
                'FEATURE_BASED_PROCESS_PLANNING.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 | (
         'FEATURE_BASED_PROCESS_PLANNING.MAPPED_ITEM' IN TYPEOF(it)) ) | (NOT
         ('FEATURE_BASED_PROCESS_PLANNING.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(['FEATURE_BASED_PROCESS_PLANNING.ELEMENTARY_SURFACE',
                 'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_SURFACE',
                 'FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE'] * TYPEOF(
                 face_geometry)) = 1;
      wr2 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                 'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
                 bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                 .edge_list | (NOT (
                 'FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE' IN TYPEOF(oe\
                 oriented_edge.edge_element))) )) = 0)) )) = 0;
      wr3 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                 'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
                 bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                 .edge_list | (NOT (SIZEOF([
                 'FEATURE_BASED_PROCESS_PLANNING.LINE',
                 'FEATURE_BASED_PROCESS_PLANNING.CONIC',
                 'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
                 'FEATURE_BASED_PROCESS_PLANNING.SURFACE_CURVE',
                 'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE'] * TYPEOF(
                 oe.edge_element\edge_curve.edge_geometry)) = 1)) )) = 0)) ))
                 = 0;
      wr4 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                 'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
                 bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
                 .edge_list | (NOT (((
                 'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(oe\
                 edge.edge_start)) AND (
                 'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(
                 oe\edge.edge_start\vertex_point.vertex_geometry))) AND ((
                 'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(oe\
                 edge.edge_end)) AND (
                 'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(
                oe\edge.edge_end\vertex_point.vertex_geometry))))) )) = 0)) ))
                 = 0;
      wr5 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
                 'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
                 bound)) ) | ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_PATH'
                 IN TYPEOF(elp_fbnds.bound)) )) = 0;
      wr6 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
                 TYPEOF(face_geometry))) OR (SIZEOF([
                 'FEATURE_BASED_PROCESS_PLANNING.LINE',
                 'FEATURE_BASED_PROCESS_PLANNING.CONIC',
                 'FEATURE_BASED_PROCESS_PLANNING.POLYLINE',
```

```
            'FEATURE_BASED_PROCESS_PLANNING.B_SPLINE_CURVE'] * TYPEOF(
            face_geometry\swept_surface.swept_curve)) = 1);
    wr7 : SIZEOF(QUERY ( vlp_fbnds <* QUERY ( bnds <* bounds | (
            'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP' IN TYPEOF(bnds
            .bound)) ) | (NOT ((
            'FEATURE_BASED_PROCESS_PLANNING.VERTEX_POINT' IN TYPEOF(
            vlp_fbnds\face_bound.bound\vertex_loop.loop_vertex)) AND (
            'FEATURE_BASED_PROCESS_PLANNING.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([
            'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP',
            'FEATURE_BASED_PROCESS_PLANNING.VERTEX_LOOP'] * TYPEOF(bnd.
            bound)) = 1)) )) = 0;
    wr9 : SIZEOF(QUERY ( elp_fbnds <* QUERY ( bnds <* bounds | (
            'FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN TYPEOF(bnds.
            bound)) ) | (NOT (SIZEOF(QUERY ( oe <* elp_fbnds.bound\path
            .edge_list | ((
            'FEATURE_BASED_PROCESS_PLANNING.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 (
            'FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(sc_ag))) ))
            = 0))) )) = 0)) )) = 0;
    wr10: ((NOT ('FEATURE_BASED_PROCESS_PLANNING.SWEPT_SURFACE' IN
            TYPEOF(face_geometry))) OR ((NOT (
            'FEATURE_BASED_PROCESS_PLANNING.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 | ('FEATURE_BASED_PROCESS_PLANNING.EDGE_LOOP' IN
            TYPEOF(bnds.bound)) ) | (NOT (SIZEOF(QUERY ( oe <*
            elp_fbnds.bound\path.edge_list | ((
            'FEATURE_BASED_PROCESS_PLANNING.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 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 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,'FEATURE_BASED_PROCESS_PLANNING.' +
             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
    wr2: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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_area
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(SELF.of_shape); wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
      ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'effective length')) )) = 1)) ))
              = 0)) )) <= 1;
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'maximum length')) )) <= 1)) ))
              = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'FEATURE_BASED_PROCESS_PLANNING.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;
    WHERE
      wr1: NOT (((
              'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_CLASS' IN
               TYPEOF(SELF.assigned_class)) AND (SELF.role.name =
              'definitional class membership')) AND (SELF.assigned_class.
              name = 'library identifier'));
      wr2: SIZEOF(QUERY ( edir <* USEDIN(SELF.assigned_class,
      'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.'
              + 'RELATED_ITEM') | (((edir.name = 'name scope') AND ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
```



```
                'EXTERNALLY_DEFINED_GENERAL_PROPERTY') IN TYPEOF(edir.
                relating_item))) AND (NOT (SIZEOF(QUERY ( gpa <* USEDIN(edir
                .relating_item,'FEATURE_BASED_PROCESS_PLANNING.' +
                'GENERAL_PROPERTY_ASSOCIATION.BASE_DEFINITION') | ((gpa.name
                = 'definitional') AND (NOT (SIZEOF(QUERY ( pdr <* USEDIN(gpa
                .derived_definition,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((pdr.
                used_representation.name = 'property value') AND (
                'FEATURE_BASED_PROCESS_PLANNING.VALUE_REPRESENTATION_ITEM'
                IN TYPEOF(pdr.used_representation.items)))
                )) >= 1))) )) = 0))) ))
                = 0;
END_ENTITY; -- applied_classification_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_reference_item;
END_ENTITY; -- applied_document_usage_constraint_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_item;
END_ENTITY; -- applied_identification_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,'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,'FEATURE_BASED_PROCESS_PLANNING.' +
               '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: ((('FEATURE_BASED_PROCESS_PLANNING.UNIFORM_CURVE' IN TYPEOF(
             SELF)) OR (
             'FEATURE_BASED_PROCESS_PLANNING.QUASI_UNIFORM_CURVE' IN
             TYPEOF(SELF))) OR (
             'FEATURE_BASED_PROCESS_PLANNING.BEZIER_CURVE'
             IN TYPEOF(SELF)))
             OR (
             'FEATURE_BASED_PROCESS_PLANNING.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: ((('FEATURE_BASED_PROCESS_PLANNING.UNIFORM_SURFACE' IN TYPEOF(
             SELF)) OR (
             'FEATURE_BASED_PROCESS_PLANNING.QUASI_UNIFORM_SURFACE' IN
             TYPEOF(SELF))) OR (
             'FEATURE_BASED_PROCESS_PLANNING.BEZIER_SURFACE' IN TYPEOF(
             SELF))) OR
             ('FEATURE_BASED_PROCESS_PLANNING.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 | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
             AND (it.name = 'orientation')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1;
    wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
               | ((sa_occ.description = 'boss height occurrence') AND (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'path feature component usage') AND ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (((
              'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (
          'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
          IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
          pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
           | ((sa_occ.description = 'circular profile occurrence')
          AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') | ((sar.description =
          'profile usage') AND
         ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (
          'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
           TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) =
          0);
  wr7 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (
          'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
          IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
          pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
           | ((sa_occ.description = 'top condition occurrence') AND (
          SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
          'FEATURE_BASED_PROCESS_PLANNING.'
          + 'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
           | ((sar.description = 'boss top usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (
          'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (
          'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
          IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
          pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
```

```
                | ((sa_occ.description = 'change in diameter occurrence')
               AND (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'taper usage') AND
            ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
               related_shape_aspect)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
               pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                | ((sa_occ.description = 'enclosed boundary occurrence')
               AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
               pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                | ((sa_occ.description = 'change in boundary occurrence')
               AND (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') | ((sar.description =
               'taper usage') AND
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (((
               'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
               related_shape_aspect)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
               IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
               pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
```

```
                 | ((sa_occ.description = 'rectangular profile occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
                ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) |
                ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE'
                IN TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) ))
                = 0);
      wr12: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
                SELF) |
               (('FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           ('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
      wr4: (NOT (SELF.description = 'planar')) OR (SIZEOF(QUERY ( pd <*
               USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              ('FEATURE_BASED_PROCESS_PLANNING.LOCATION_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
      wr5: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
               USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
               'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
      wr6: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') | (((sar.description =
               'boss top usage') AND (sar.name IN ['boss height start',
               'boss height end'])) AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
               = 'top condition occurrence') AND (
```

```
               'FEATURE_BASED_PROCESS_PLANNING.BOSS' IN TYPEOF(fcr.
               related_shape_aspect.of_shape.definition))) AND (
               'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP' IN TYPEOF(fcr.
               relating_shape_aspect))) )) >= 1;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) <= 1)) )) = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'top radius')) )) <= 1)) ))
               = 0)) )) = 0;
END_ENTITY; -- boss_top

ENTITY bounded_curve
  SUPERTYPE OF (ONEOF (polyline,b_spline_curve,composite_curve))
  SUBTYPE OF (curve);
END_ENTITY; -- bounded_curve

ENTITY bounded_surface
  SUPERTYPE OF (b_spline_surface)
  SUBTYPE OF (surface);
END_ENTITY; -- bounded_surface

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 chamfer
  SUBTYPE OF (transition_feature);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATING_SHAPE_ASPECT') |
         ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
             IN TYPEOF(sar)) ) | (((
             'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN TYPEOF(
             fcr.related_shape_aspect)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.CHAMFER' IN TYPEOF(fcr.
             relating_shape_aspect))) AND (fcr.related_shape_aspect.
             description = 'first offset')) )) = 1;
    wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATING_SHAPE_ASPECT') |
```



```
            ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) ) | (((
              'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET' IN TYPEOF(
              fcr.related_shape_aspect)) AND (
              'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```

```
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
    (((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'offset amount')) OR ((
             SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATED_SHAPE_ASPECT') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'FEATURE_COMPONENT_RELATIONSHIP') IN TYPEOF(sar)) ) | ((
             'FEATURE_BASED_PROCESS_PLANNING.CHAMFER' IN TYPEOF(sdr.
             relating_shape_aspect)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
             IN TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```



```
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((
            'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
            AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (
            'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | (
            SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATING_SHAPE_ASPECT') |
         ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar)) ) | (
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT' IN TYPEOF(sdr.
            related_shape_aspect)) )) = 1) )) <= 3)) )) = 0;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
```

```
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'diameter')) )) <= 1)) )) =
              0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'base feature rotation')) ))
             <= 1)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
            ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.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,
```

```
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
           * TYPEOF(it)) = 2) AND (it.name = 'angular spacing')) )) = 1)) ))
             = 0)) )) = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
              'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
              AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
              'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
              'FEATURE_BASED_PROCESS_PLANNING.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 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_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: 'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | ((
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
             'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
             TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
             component_relationships |
          (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
             IN TYPEOF(sar)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(sar.
             related_shape_aspect))) )) = 2)) )) = 1)) )) = 1;
    wr3: (NOT (SELF\characterized_object.description = 'countersunk'))
             OR (SIZEOF(QUERY ( pds <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | ((
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
             'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
             TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
             component_relationships | ((
             'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(sar.
             related_shape_aspect)) AND (NOT (SIZEOF(QUERY ( pds <*
             QUERY ( pd <* USEDIN(sar.related_shape_aspect,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
             sa_occ.description = 'change in diameter occurrence') AND (
             SIZEOF(QUERY ( fcr2 <* QUERY ( sar2 <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.'
              + 'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT')
              | ((sar2.description = 'taper usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
             IN TYPEOF(sar2))) ) | (
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               csa.name = 'compound feature in solid') AND (
               'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa))) )) = 1)) )) = 1;
      wr2: SIZEOF(QUERY ( pds <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | (
               'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa)) )) = 1)) )) = 1;
      wr3: SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa)) AND (SIZEOF(QUERY ( fcr <* csa.
               component_relationships | (NOT
             ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(fcr))) )) = 0)) )) = 1)) )) = 1;
      wr4: SIZEOF(QUERY ( pds <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | (
               'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa)) )) = 1)) )) = 1;
      wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
               component_relationships | (
               'FEATURE_BASED_PROCESS_PLANNING.THREAD' IN TYPEOF(sar.
               related_shape_aspect)) )) = 0)) )) = 1)) )) = 1;
      wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
```



```
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | ((
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
              'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
              TYPEOF(csa)) AND (SIZEOF(QUERY ( sar <* csa.
              component_relationships | ((
              'FEATURE_BASED_PROCESS_PLANNING.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_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 conversion_based_unit
  SUBTYPE OF (named_unit);
    name              : label;
    conversion_factor : measure_with_unit;
END_ENTITY; -- conversion_based_unit
```

```
ENTITY curve
  SUPERTYPE OF (ONEOF (line,conic,pcurve,surface_curve))
  SUBTYPE OF (geometric_representation_item);
END_ENTITY; -- curve

ENTITY cylindrical_shape_representation
  SUBTYPE OF (shape_representation_with_parameters);
  WHERE
    wr1: SIZEOF(SELF.items) = 3;
    wr2: SIZEOF(QUERY ( it <* SELF.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
             AND (it.name = 'orientation')) )) = 1;
    wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
    wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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 (('FEATURE_BASED_PROCESS_PLANNING.' +
             '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 (ONEOF (calendar_date,ordinal_date,
      week_of_year_and_day_date));
    year_component : year_number;
END_ENTITY; -- date

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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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) * [
             'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATING_SHAPE_ASPECT')) | (NOT (
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATING_SHAPE_ASPECT')) | (NOT (
             'FEATURE_BASED_PROCESS_PLANNING.DATUM' IN TYPEOF(sar.
             related_shape_aspect))) )) = 0;
    wr2: SELF.product_definitional = TRUE;
END_ENTITY; -- datum_target

ENTITY definitional_representation
  SUBTYPE OF (representation);
```

```
    WHERE
      wr1: 'FEATURE_BASED_PROCESS_PLANNING.PARAMETRIC_REPRESENTATION_CONTEXT'
               IN TYPEOF(SELF\representation.context_of_items);
  END_ENTITY; -- definitional_representation

  ENTITY degenerate_toroidal_surface
    SUBTYPE OF (toroidal_surface);
      select_outer : BOOLEAN;
    WHERE
      wr1: major_radius < minor_radius;
  END_ENTITY; -- degenerate_toroidal_surface

  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,'FEATURE_BASED_PROCESS_PLANNING.' +
               '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 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 (
             'FEATURE_BASED_PROCESS_PLANNING.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 (characterized_object, document);
```

```
  WHERE
    wr1: (SIZEOF(QUERY ( adr <* QUERY ( dr <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_REFERENCE.ASSIGNED_DOCUMENT')
             | (
            'FEATURE_BASED_PROCESS_PLANNING.APPLIED_DOCUMENT_REFERENCE'
            IN TYPEOF(dr)) ) |
    ('FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION'
            IN TYPEOF(adr.items)) )) = 1) OR (SIZEOF(QUERY ( duc <*
            USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_USAGE_CONSTRAINT.SOURCE')
             | (NOT (SIZEOF(QUERY ( aduc <* QUERY ( duca <* USEDIN(duc,
     'FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.'
            + 'ASSIGNED_DOCUMENT_USAGE') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT') IN TYPEOF(
            duca)) ) |
    ('FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION'
            IN TYPEOF(aduc.items)) )) = 1)) )) = 0);
    wr2: SIZEOF(QUERY ( drt <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'DOCUMENT_REPRESENTATION_TYPE.REPRESENTED_DOCUMENT') | (drt.
            name = 'physical') )) = 1;
END_ENTITY; -- document_file

ENTITY document_reference
  ABSTRACT SUPERTYPE;
    assigned_document : document;
    source            : label;
  DERIVE
    role : object_role := get_role(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
            '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 document_with_class
  SUBTYPE OF (document);
    class : identifier;
END_ENTITY; -- document_with_class

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_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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
          ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
          ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((SIZEOF(
          ['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
          TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
          = 0)) )) = 0);
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
          IN TYPEOF(pdr.used_representation)) AND (pdr.
          used_representation.name = 'edge round face')) )) <= 1)) ))
          = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```



```
                'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'first face shape')) )) <= 1)) ))
                = 0;
      wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'second face shape')) )) <= 1)) ))
                = 0;
  END_ENTITY; -- edge_round

  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 executed_action
    SUBTYPE OF (action);
  END_ENTITY; -- executed_action

  ENTITY expanded_uncertainty
    SUBTYPE OF (standard_uncertainty);
      coverage_factor : REAL;
  END_ENTITY; -- expanded_uncertainty

  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,'FEATURE_BASED_PROCESS_PLANNING.' +
                'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  END_ENTITY; -- external_source

  ENTITY externally_defined_class
    SUBTYPE OF (externally_defined_item, class);
    WHERE
      wr1: 'FEATURE_BASED_PROCESS_PLANNING.KNOWN_SOURCE' IN TYPEOF(SELF.
                source);
```

```
    wr2: SELF.source.name = 'ISO 13584 library';
WR3: ( SIZEOF ( QUERY ( aoa <*
     USEDIN ( SELF.source,
     'FEATURE_BASED_PROCESS_PLANNING.'+
     'FEATURE_BASED_PP_ORGANIZATION_ASSIGNMENT.ITEMS' )|
     aoa. role.name = 'library supplier' ))=1);

wr4: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.' +
       'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') |
       (('FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.APPLIED_DOCUMENT_REFERENCE.ITEMS')
              | (adr.assigned_document.description =
             'externally defined size 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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         '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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         '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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
         IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
         pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
          | ((sa_occ.description = 'partial area occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'applied area usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | (
         'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
         .relating_shape_aspect)) )) = 1)) )) <= 1)) )) = 0);
wr6 : (NOT (SELF\characterized_object.description = 'marking')) OR (
         SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
         pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items |
  (('FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
         pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
```

```
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
   ((('FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
    (('FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
     (('FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
       (('FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
```



```
        (('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
    ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
  ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *
                TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
                1)) )) = 0)) )) = 0);
    wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
               IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
               = 0)) )) = 0;
    wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
    ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (
            'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
            IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
            pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
             | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
            + 'RELATED_SHAPE_ASPECT') | ((sar.description =
            'applied shape') AND
         ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
            IN TYPEOF(sar))) ) | (
            'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            '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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items |
  ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
            pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
```



```
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'FEATURE_BASED_PROCESS_PLANNING.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'FEATURE_BASED_PROCESS_PLANNING.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'FEATURE_BASED_PROCESS_PLANNING.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'FEATURE_BASED_PROCESS_PLANNING.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'FEATURE_BASED_PROCESS_PLANNING.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'FEATURE_BASED_PROCESS_PLANNING.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
            | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
           pd,'FEATURE_BASED_PROCESS_PLANNING.' +
           'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
           'FEATURE_BASED_PROCESS_PLANNING.' +
           'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
           used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
           impl_rep.used_representation.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```

```
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
        ((('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
       ((('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
           'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.KNOWN_SOURCE' IN TYPEOF(SELF.
             source);
    wr2: SELF.source.name = 'ISO 13584 library';
    wr3: SIZEOF(QUERY ( aoa <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT.ITEMS') | (((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'LIBRARY_CLASS_VERSION_ASSIGNMENT') IN TYPEOF(aoa)) AND (aoa
             .role.name = 'property version')) )) = 1;
    wr4: SIZEOF(QUERY ( ap <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'EXTERNALLY_DEFINED_ITEM_RELATIONSHIP.RELATING_ITEM') | ((ap
             .name = 'name scope') AND (
             'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'APPLIED_CLASSIFICATION_ASSIGNMENT.ITEMS') | (
             'FEATURE_BASED_PROCESS_PLANNING.CLASSIFICATION_ASSIGNMENT'
             IN TYPEOF(adr)) )) = 1;
    wr2: SIZEOF(QUERY ( adr <* SELF.items | (
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(adr)) ))
             = 1;
    wr3: SIZEOF(QUERY ( adr <* SELF.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(adr))
             AND ('FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN
             TYPEOF(adr.location))) )) = 1;
END_ENTITY; -- externally_defined_representation_with_parameters

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 | (
             'FEATURE_BASED_PROCESS_PLANNING.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 ((
             'FEATURE_BASED_PROCESS_PLANNING.FACE_SURFACE' IN TYPEOF(it))
             OR ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_FACE' IN
             TYPEOF(it)))) )) = 0;
END_ENTITY; -- face_shape_representation

ENTITY face_shape_representation_relationship
  SUBTYPE OF (representation_relationship);
  WHERE
    wr1: 'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
             TYPEOF(SELF.rep_1);
    wr2: 'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
             TYPEOF(SELF.rep_2);
END_ENTITY; -- face_shape_representation_relationship

ENTITY face_surface
  SUBTYPE OF (face, geometric_representation_item);
    face_geometry : surface;
    same_sense    : BOOLEAN;
  WHERE
    wr1: NOT ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_SURFACE' IN
             TYPEOF(face_geometry));
END_ENTITY; -- face_surface

ENTITY feature_based_pp_action_assignment
  SUBTYPE OF (action_assignment);
    items : SET [1:?] OF feature_based_pp_action_item;
END_ENTITY; -- feature_based_pp_action_assignment

ENTITY feature_based_pp_action_request_assignment
  SUBTYPE OF (action_request_assignment);
    items : SET [1:?] OF feature_based_pp_action_request_item;
END_ENTITY; -- feature_based_pp_action_request_assignment

ENTITY feature_based_pp_approval_assignment
  SUBTYPE OF (approval_assignment);
    items : SET [1:?] OF feature_based_pp_approved_item;
END_ENTITY; -- feature_based_pp_approval_assignment

ENTITY feature_based_pp_date_assignment
```



```
  SUBTYPE OF (date_assignment);
    items : SET [1:?] OF feature_based_pp_dated_item;
END_ENTITY; -- feature_based_pp_date_assignment

ENTITY feature_based_pp_organization_assignment
  SUBTYPE OF (organization_assignment);
    items : SET [1:?] OF feature_based_pp_organization_item;
END_ENTITY; -- feature_based_pp_organization_assignment

ENTITY feature_based_pp_person_and_organization_assignment
  SUBTYPE OF (person_and_organization_assignment);
    items : SET [1:?] OF feature_based_pp_person_and_organization_item;
END_ENTITY; -- feature_based_pp_person_and_organization_assignment

ENTITY feature_based_pp_security_classification_assignment
  SUBTYPE OF (security_classification_assignment);
    items : SET [1:?] OF feature_based_pp_classified_item;
END_ENTITY; -- feature_based_pp_security_classification_assignment

ENTITY feature_component_definition
  SUBTYPE OF (characterized_object);
  WHERE
    wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.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([
             'FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT',
             'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
             'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE',
             'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN'] * TYPEOF(
             SELF.relating_shape_aspect)) = 1) OR (
             'FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION' IN
             TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
             ('FEATURE_BASED_PROCESS_PLANNING.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) |
     ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) )) <= 1;
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
```

```
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
                AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
      wr3: SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.BOSS',
                'FEATURE_BASED_PROCESS_PLANNING.TURNED_KNURL',
                'FEATURE_BASED_PROCESS_PLANNING.THREAD',
                'FEATURE_BASED_PROCESS_PLANNING.GEAR',
                'FEATURE_BASED_PROCESS_PLANNING.MARKING',
                'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP',
                'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE',
                'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.POCKET',
                'FEATURE_BASED_PROCESS_PLANNING.REMOVAL_VOLUME',
                'FEATURE_BASED_PROCESS_PLANNING.REVOLVED_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.OUTER_ROUND',
                'FEATURE_BASED_PROCESS_PLANNING.FLAT_FACE',
                'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION',
                'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_END',
                'FEATURE_BASED_PROCESS_PLANNING.SLOT',
                'FEATURE_BASED_PROCESS_PLANNING.SPHERICAL_CAP',
                'FEATURE_BASED_PROCESS_PLANNING.STEP',
                'FEATURE_BASED_PROCESS_PLANNING.COMPOUND_FEATURE',
                'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
    'FEATURE_BASED_PROCESS_PLANNING.EXTERNALLY_DEFINED_FEATURE_DEFINITION']
                * TYPEOF(SELF)) <= 1;
      wr4: (NOT (SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE',
                'FEATURE_BASED_PROCESS_PLANNING.BOSS',
                'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.REMOVAL_VOLUME',
                'FEATURE_BASED_PROCESS_PLANNING.FLAT_FACE',
                'FEATURE_BASED_PROCESS_PLANNING.POCKET',
                'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION',
                'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP',
                'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_END',
                'FEATURE_BASED_PROCESS_PLANNING.SLOT',
                'FEATURE_BASED_PROCESS_PLANNING.STEP'] * TYPEOF(SELF)) = 1))
                OR (SIZEOF(QUERY ( pdr <*
                get_property_definition_representations(SELF) |
               (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'maximum feature limit')) )) >= 0);
  END_ENTITY; -- feature_definition

  ENTITY feature_pattern
    SUBTYPE OF (replicate_feature);
    WHERE
      wr1:            SIZEOF(QUERY            (            pd            <*
USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```



```
              used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
              used_representation.items | (NOT (
           'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(srwp_i))) ))
               > 0)) )) > 0) )) = 0;
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((
              'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
              AND (it.name = 'base feature placement')) )) > 1)) )) = 0)) ))
              = 0;
  END_ENTITY; -- feature_pattern

  ENTITY fillet
    SUBTYPE OF (transition_feature);
    WHERE
      wr1: (NOT (SELF\shape_aspect.description = 'constant radius')) OR (
              SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.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,
```

```
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
             TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) <= 1)) ))
             = 0)) )) = 0);
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
             'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'fillet face')) )) = 1)) )) = 0;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
             'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'first face shape')) )) = 1)) ))
             = 0;
    wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
             'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'second face shape')) )) = 1)) ))
             = 0;
END_ENTITY; -- fillet

ENTITY flat_face
  SUBTYPE OF (feature_definition);
```



```
WHERE
  wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('FEATURE_BASED_PROCESS_PLANNING.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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'course of travel occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'path feature component usage') AND
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | (((
         'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'removal boundary occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'profile usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | ((
         'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE' IN TYPEOF(
         sdr.relating_shape_aspect)) AND (sdr.name =
         'removal boundary')) )) = 1)) )) = 1)) )) = 0;
  wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'enclosed boundary occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'profile usage') AND
       ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
```

```
            'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
            'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
            'FEATURE_BASED_PROCESS_PLANNING.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) |
          (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
            IN TYPEOF(pdr.used_representation)) AND (pdr.
            used_representation.name = 'maximum feature limit')) )) >= 0;
    wr6: SIZEOF(QUERY ( pds <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | ((
            'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
            ('FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
            TYPEOF(csa)) AND (csa.name = 'uncut volume')) AND (SIZEOF(
            QUERY ( sar <* csa.component_relationships |
         (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar)) AND (SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.BOSS',
            'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION'] * TYPEOF(sar.
            related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
            used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
            impl_rep.used_representation.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.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 (('FEATURE_BASED_PROCESS_PLANNING.' +
            '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ((('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
   ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')  |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((
             SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((
             SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')  |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
             IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
             QUERY ( it <* impl_rep.used_representation.items | ((
             SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')  |
  ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
```



```
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items |
   ((('FEATURE_BASED_PROCESS_PLANNING.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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
  ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items |
   ((('FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
   ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((
          SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((
          SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')  |
```

```
 ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((
          SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items |
    ((('FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((
          SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
          IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
          QUERY ( it <* impl_rep.used_representation.items | ((
          SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'root angle')) )) = 1)) ))
          = 0)) )) = 0);
wr16: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (
          'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
```

```
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT')
                 | ((sar.description = 'applied shape') AND
                ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.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,(
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'GENERAL_PROPERTY_ASSOCIATION.') + 'DERIVED_DEFINITION')) =
             1;
    wr2: derived_definition.name = base_definition.name;
END_ENTITY; -- general_property_association

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,vertex_point,solid_model))
  SUBTYPE OF (representation_item);
  DERIVE
    dim : dimension_count := dimension_of(SELF);
  WHERE
    wr1: SIZEOF(QUERY ( using_rep <* using_representations(SELF) | (NOT
         ('FEATURE_BASED_PROCESS_PLANNING.GEOMETRIC_REPRESENTATION_CONTEXT'
             IN TYPEOF(using_rep.context_of_items))) )) = 0;
END_ENTITY; -- geometric_representation_item

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 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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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 : 'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
              + '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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0);
    wr5 : (NOT (SELF.description = 'flat')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```

```
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATING_SHAPE_ASPECT') | ((sar.description =
             'hole bottom usage') AND
         ('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
      'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
          * TYPEOF(it)) = 2) AND (it.name = 'tip angle')) )) = 1)) ))
          = 0)) )) = 0);
    wr14: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATING_SHAPE_ASPECT') | ((sar.description =
          'hole bottom usage') AND
      ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
          IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
          description = 'bottom condition occurrence') AND (
          'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
          related_shape_aspect.of_shape.definition))) AND (
          'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM' IN TYPEOF(fcr.
          relating_shape_aspect))) )) >= 1;
    wr15: (NOT (SELF.description = 'flat with taper')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.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 instanced_feature
  SUBTYPE OF (feature_definition, shape_aspect);
  WHERE
    wr1: 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION' IN TYPEOF(
             SELF.of_shape.definition);
    wr2: SELF.product_definitional;
END_ENTITY; -- instanced_feature

ENTITY known_source
  SUBTYPE OF (external_source, pre_defined_item);
  WHERE
    wr1: SELF\pre_defined_item.name = 'ISO 13584 library';
    wr2: SIZEOF(QUERY ( oa <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.'+
    'FEATURE_BASED_PP_ORGANIZATION_ASSIGNMENT.ITEMS')|
    ((NOT ('FEATURE_BASED_PROCESS_PLANNING.'+
    'ORGANIZATION_ASSIGNMENT'
    IN TYPEOF(oa))) AND (oa.role.name = 'library supplier')) ))
    = 0;
END_ENTITY; -- known_source

ENTITY length_measure_with_unit
  SUBTYPE OF (measure_with_unit);
  WHERE
    wr1: 'FEATURE_BASED_PROCESS_PLANNING.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 (('FEATURE_BASED_PROCESS_PLANNING.' +
                '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,'FEATURE_BASED_PROCESS_PLANNING.' +
                '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: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0;
```

```
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
             AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'profile length')) )) = 1)) ))
             = 0)) )) = 0;
END_ENTITY; -- linear_profile

ENTITY location_shape_representation
  SUBTYPE OF (shape_representation);
  WHERE
    wr1: SIZEOF(SELF.items) = 1;
    wr2: SIZEOF(QUERY ( it <* SELF.items | (NOT (
             'FEATURE_BASED_PROCESS_PLANNING.POINT' IN TYPEOF(it))) )) =
             0;
END_ENTITY; -- location_shape_representation

ENTITY loop
  SUPERTYPE OF (ONEOF (vertex_loop,edge_loop))
  SUBTYPE OF (topological_representation_item);
END_ENTITY; -- loop

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 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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
      (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'marking text')) )) = 1)) )) =
              0)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
       (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (it.name = 'special instructions')) )) <=
              1)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
```

```
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
           (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'font name')) )) <= 1)) )) = 0)) ))
               = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'character height')) )) <=
             1)) )) = 0)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'character spacing')) )) <=
              1)) )) = 0)) )) = 0;
    wr7: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
              TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | (
             SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATED_SHAPE_ASPECT') | ((sar.description =
             'applied shape') AND
             ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
             IN TYPEOF(sar))) ) | (
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT' IN TYPEOF(sdr.
             relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
END_ENTITY; -- marking

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: ('FEATURE_BASED_PROCESS_PLANNING.CHARACTERIZED_OBJECT' IN
               TYPEOF(SELF\property_definition.definition)) OR (SIZEOF(
               bag_to_set(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) -
               QUERY ( temp <* bag_to_set(USEDIN(SELF,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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 | (
               'FEATURE_BASED_PROCESS_PLANNING.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,
        plane_angle_measure_with_unit,solid_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,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') |
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
```

```
              IN TYPEOF(sar)) ) | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
              'FEATURE_BASED_PROCESS_PLANNING.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,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATING_SHAPE_ASPECT') |
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar)) ) | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
              'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
              TYPEOF(fcr.related_shape_aspect.of_shape.definition)) = 1)
              AND (fcr.description = 'base pattern')) )) = 1;
      wr3: SIZEOF(QUERY ( sar <* USEDIN(SELF,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATING_SHAPE_ASPECT') | (SIZEOF(QUERY ( msar <* USEDIN(
              sar.related_shape_aspect,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | (((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.PATTERN_OFFSET_MEMBERSHIP',
              'FEATURE_BASED_PROCESS_PLANNING.PATTERN_OMIT_MEMBERSHIP'] *
              TYPEOF(sar)) = 1) AND (sar.description = 'modified pattern'))
              AND (sar :<>: msar)) )) >= 1) )) = 0;
  END_ENTITY; -- modified_pattern

  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,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: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
      wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         '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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         '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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((
         'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
         AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items |
     ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
         IN TYPEOF(it)) AND (
         'FEATURE_BASED_PROCESS_PLANNING.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,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
```

```
              'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.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 | ((
              'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
              AND (it.name = 'orientation')) )) = 1;
      wr3: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1;
      wr4: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) = 1;
      wr5: SIZEOF(QUERY ( it <* SELF.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name IN ['circumscribed diameter',
              'diameter across flats'])) )) = 1;
      wr6: SIZEOF(QUERY ( it <* SELF.items |
          ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (
              'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
              measure_with_unit.value_component))) AND (it.name =
              'number of sides')) )) = 1;
  END_ENTITY; -- ngon_shape_representation

  ENTITY object_role;
      name        : label;
      description : OPTIONAL text;
  END_ENTITY; -- object_role

  ENTITY open_path_profile
    SUBTYPE OF (shape_aspect);
    WHERE
      wr1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
              IN TYPEOF(SELF.of_shape.definition);
```

```
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
             AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (
             'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0;
    wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           (('FEATURE_BASED_PROCESS_PLANNING.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 ordered_part
  SUBTYPE OF (action_assignment, characterized_object);
    items : SET [1:?] OF feature_based_pp_ordered_item;
  WHERE
    wr1: SIZEOF(USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION'))
             = 1;
```

```
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
             + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
             = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
             SIZEOF(items) = 1)) )) = 0)) )) = 0;
    wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
             SIZEOF(QUERY ( it <* pdr.used_representation.items |
           (('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'quantity')) )) = 1)) )) = 0)) ))
             = 0;
END_ENTITY; -- ordered_part

ENTITY ordinal_date
  SUBTYPE OF (date);
    day_component : day_in_year_number;
  WHERE
    wr1: ((NOT leap_year(SELF.year_component)) AND ((1 <= day_component)
             AND (day_component <= 365))) OR (leap_year(SELF.
             year_component) AND ((1 <= day_component) AND (day_component
             <= 366)));
END_ENTITY; -- ordinal_date

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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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 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 ('FEATURE_BASED_PROCESS_PLANNING.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 ('FEATURE_BASED_PROCESS_PLANNING.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 ('FEATURE_BASED_PROCESS_PLANNING.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 ('FEATURE_BASED_PROCESS_PLANNING.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 ('FEATURE_BASED_PROCESS_PLANNING.ORIENTED_PATH' IN TYPEOF(
             SELF.path_element));
END_ENTITY; -- oriented_path

ENTITY outer_round
  SUBTYPE OF (feature_definition);
  WHERE
    wr1: (NOT (SELF\characterized_object.description = 'outer diameter'))
             OR (SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    (('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
            TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
             = 0);
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```



```
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
         'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'v-shape boundary occurrence') AND (
         SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'profile usage') AND
       ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
         IN TYPEOF(sar))) ) | ((
         'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (
         'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
          TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
         sa_occ.description = 'reduced size occurrence') AND (SIZEOF(
         QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
         'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
         + 'RELATED_SHAPE_ASPECT') | ((sar.description =
         'taper usage') AND
       ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
         IN TYPEOF(sar))) ) | (((
         'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(sdr.
         relating_shape_aspect)) AND (
         'FEATURE_BASED_PROCESS_PLANNING.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,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```

```
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | (sa_occ.description = 'boundary occurrence') )) = 1)) ))
                = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
```



```
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
 | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile usage') AND (('FEATURE_BASED_PROCESS_PLANNING.' +
'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | ((
SIZEOF([
'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
'FEATURE_BASED_PROCESS_PLANNING.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,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
pds,'FEATURE_BASED_PROCESS_PLANNING.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,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
 | (
'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
 | (NOT (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
+ 'RELATED_SHAPE_ASPECT') | ((sar.description =
'profile floor usage') AND
('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
IN TYPEOF(sar))) ) | ((
'FEATURE_BASED_PROCESS_PLANNING.PROFILE_FLOOR' IN TYPEOF(
sdr.relating_shape_aspect)) AND (
'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE' IN TYPEOF(
sdr.related_shape_aspect.of_shape.definition))) )) = 1)) ))
= 0)) )) = 0);
wr5: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')|
('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
 IN TYPEOF(pd)) ) |
(NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
'FEATURE_BASED_PROCESS_PLANNING.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']) AND
          (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
          ((sar.description = 'path feature component usage') AND
          (sar.name = 'profile swept shape') AND
          ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
          IN TYPEOF(sar))) ) |
          (('FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
          TYPEOF(sdr.relating_shape_aspect)) AND
          (sdr.relating_shape_aspect.description = 'linear')
          )  )) = 1)) )) =1)) )) = 0;
wr6 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
               | (sa_occ.description = 'complex boundary occurrence') ))
              = 1)) )) = 0)) OR (SIZEOF(QUERY ( pds <* QUERY ( pd <*
              USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
               | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
              TYPEOF(sdr.relating_shape_aspect)) = 1) )) = 1) )) = 1)) ))
              = 0);
      wr7 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
```



```
         | (sa_occ.description =
        'partial circular boundary occurrence') )) = 1)) )) = 0))
        OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
         | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
         | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
        IN TYPEOF(sar))) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE'
        IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
         0);
wr8 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
         | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
         | (sa_occ.description =
        'closed circular boundary occurrence') )) = 1)) )) = 0)) OR
         (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
         | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
         | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND
       ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
        IN TYPEOF(sar))) ) | (
        'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
         TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
wr9 : (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
         | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
         | (sa_occ.description =
        'open rectangular boundary occurrence') )) = 1)) )) = 0))
        OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
         | (
        'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
        IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
        pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
         | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
        + 'RELATED_SHAPE_ASPECT') | ((sar.description =
        'profile usage') AND
```

```
            ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
             IN TYPEOF(sar))) ) | (
             'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE' IN
             TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) = 0);
    wr10: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
             IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
             pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
              | (sa_occ.description =
             'closed rectangular boundary occurrence') )) = 1)) )) = 0))
             OR (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
             IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
             pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
              | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
             + 'RELATED_SHAPE_ASPECT') | ((sar.description =
             'profile usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
             IN TYPEOF(sar))) ) |
              ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE'
             IN TYPEOF(sdr.relating_shape_aspect)) )) = 1) )) = 1)) )) =
              0);
    wr11: (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (
             'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
             IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
             pds,'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
        (('FEATURE_BASED_PROCESS_PLANNING.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) |
          (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
             IN TYPEOF(pdr.used_representation)) AND (pdr.
             used_representation.name = 'maximum feature limit')) )) >=
             0;
END_ENTITY; -- outside_profile
```



```
ENTITY parabola
  SUBTYPE OF (conic);
    focal_dist : length_measure;
  WHERE
    wr1: focal_dist <> 0;
END_ENTITY; -- parabola

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 partial_circular_profile
  SUBTYPE OF (shape_aspect);
  WHERE
    wr1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
             IN TYPEOF(SELF.of_shape.definition);
    wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) )) = 1)) )) = 0;
    wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
             AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
```

```
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
               = 0;
      wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'sweep angle')) )) = 1)) ))
               = 0)) )) = 0;
      wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.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 : 'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0);
      wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((
          'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
          AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr5 : (NOT (SELF.description = 'partial circular')) OR (SIZEOF(
          QUERY ( pd <* USEDIN(SELF,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          '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,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
         'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
          'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
       'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          '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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items |
((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          '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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
          pd,'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
          used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
          impl_rep.used_representation.items | ((SIZEOF([
        'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
          'FEATURE_BASED_PROCESS_PLANNING.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,
    'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
          IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
wr13: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
          USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
           | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
          'FEATURE_BASED_PROCESS_PLANNING.' +
          'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
          'FEATURE_BASED_PROCESS_PLANNING.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([
               'FEATURE_BASED_PROCESS_PLANNING.BOUNDED_CURVE',
               'FEATURE_BASED_PROCESS_PLANNING.EDGE_CURVE',
               'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
           + 'RELATING_SHAPE_ASPECT') |
          (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
           IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
           QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.
           related_shape_aspect.of_shape,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
                'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
                TYPEOF(pdr.definition)) = 1) )) = 0) )) = 0;
wr2 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
           related_shape_aspect,
           'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
           + 'RELATED_SHAPE_ASPECT') |
            (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
            IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description
           = 'modified pattern') AND (
           'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
           TYPEOF(fcr.relating_shape_aspect))) )) >= 1;
wr3 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
           related_shape_aspect,
           'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
          + 'RELATED_SHAPE_ASPECT') |
         (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
         IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
         'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN
         TYPEOF(fcr.relating_shape_aspect)) AND (NOT (SIZEOF(
                QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
                relating_shape_aspect,'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
                SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
                'FEATURE_BASED_PROCESS_PLANNING.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 ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
                + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 2)) ))
                = 0);
wr5 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
                + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
                = 0);
wr6 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
                SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
wr7 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
             ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
                \measure_with_unit.value_component))) AND (it.name =
                'index number')) )) = 1)) )) = 0)) )) = 0);
wr8 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
            (('FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT'
                IN TYPEOF(it)) AND (it.name = 'offset')) )) = 1)) )) = 0)) ))
                = 0);
wr9 : (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
                TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
```



```
              (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(impl_rep.
             used_representation.items) = 3)) )) = 0)) )) = 0);
wr10: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
             TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
              (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
         ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
             \measure_with_unit.value_component))) AND (it.name =
             'row index')) )) = 1)) )) = 0)) )) = 0);
wr11: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
             TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
              (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
       'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
             IN TYPEOF(it)) AND (
             'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it
             \measure_with_unit.value_component))) AND (it.name =
             'column index')) )) = 1)) )) = 0)) )) = 0);
wr12: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
             TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
              (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
              | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
             pd,'FEATURE_BASED_PROCESS_PLANNING.' +
             'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
             'FEATURE_BASED_PROCESS_PLANNING.' +
             'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
             used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
             impl_rep.used_representation.items | ((
             'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'
             IN TYPEOF(it)) AND (it.name = 'offset distance')) )) = 1)) ))
             = 0)) )) = 0);
wr13: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
             TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
```

```
                 (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           (('FEATURE_BASED_PROCESS_PLANNING.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,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') |
          (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | (SIZEOF(
               QUERY ( pdr <* QUERY ( pd <* USEDIN(fcr.related_shape_aspect
               .of_shape,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
               'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN'] *
               TYPEOF(pdr.definition)) = 1) )) = 0) )) = 0;
wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
               related_shape_aspect,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') |
         (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((fcr.description =
                'modified pattern') AND (
               'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN TYPEOF(
               fcr.relating_shape_aspect))) )) >= 1;
wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF.
               related_shape_aspect,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') |
          (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (sar :<>: SELF)) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.MODIFIED_PATTERN' IN TYPEOF(
               fcr.relating_shape_aspect)) AND (NOT (SIZEOF(
               QUERY ( modfcr <* QUERY ( modsar <* USEDIN(fcr.
               relating_shape_aspect,'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_ASPECT_RELATIONSHIP.RELATING_SHAPE_ASPECT') | ((
               SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN',
               'FEATURE_BASED_PROCESS_PLANNING.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,
```



```
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
               + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION')) = 1)) ))
               = 0;
wr5: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
               SIZEOF(pdr.used_representation.items) = 1)) )) = 0)) )) = 0);
wr6: (NOT ('FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
            ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
               IN TYPEOF(it)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
               measure_with_unit.value_component))) AND (it.name =
               'index number')) )) = 1)) )) = 0)) )) = 0);
wr7: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (NOT (
               SIZEOF(pdr.used_representation.items) = 2)) )) = 0)) )) = 0);
wr8: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
               IN TYPEOF(it)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.COUNT_MEASURE' IN TYPEOF(it\
               measure_with_unit.value_component))) AND (it.name =
               'row index')) )) = 1)) )) = 0)) )) = 0);
wr9: (NOT ('FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_PATTERN' IN
               TYPEOF(SELF.relating_shape_aspect.of_shape.definition))) OR
               (SIZEOF(QUERY ( pd <* USEDIN(SELF.related_shape_aspect,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
```

```
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
               IN TYPEOF(it)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.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 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,'FEATURE_BASED_PROCESS_PLANNING.' +
               'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
  END_ENTITY; -- person_and_organization
```



```
ENTITY person_and_organization_assignment
  ABSTRACT SUPERTYPE;
    assigned_person_and_organization : person_and_organization;
    role                             : person_and_organization_role;
END_ENTITY; -- person_and_organization_assignment

ENTITY person_and_organization_role;
    name : label;
  DERIVE
    description : text := get_description_value(SELF);
  WHERE
    wr1: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
             'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
END_ENTITY; -- person_and_organization_role

ENTITY personal_address
  SUBTYPE OF (address);
    people      : SET [1:?] OF person;
    description : OPTIONAL text;
END_ENTITY; -- personal_address

ENTITY placed_datum_target_feature
  SUBTYPE OF (datum_target);
  WHERE
    wr1 : SELF.description IN ['point','line','rectangle','circle'];
    wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) )) = 1)) )) = 0;
    wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              dtm_rep.used_representation.items | ((
              'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
              AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
    wr4 : (NOT (SELF.description = 'point')) OR (SIZEOF(QUERY ( pd <*
              USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(dtm_rep.
              used_representation.items) = 1)) )) = 0)) )) = 0);
    wr5 : (NOT (SELF.description IN ['line','circle'])) OR (SIZEOF(
              QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```

```
          | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(dtm_rep.
         used_representation.items) = 2)) )) = 0)) )) = 0);
wr6 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
         QUERY ( pd <* USEDIN(SELF,
      'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(dtm_rep.
         used_representation.items) = 3)) )) = 0)) )) = 0);
wr7 : (NOT (SELF.description = 'circle')) OR (SIZEOF(QUERY ( pd <*
         USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         dtm_rep.used_representation.items | ((SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
         'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
         * TYPEOF(it)) = 2) AND (it.name = 'target diameter')) )) =
         1)) )) = 0)) )) = 0);
wr8 : (NOT (SELF.description = 'line')) OR (SIZEOF(QUERY ( pd <*
         USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         dtm_rep.used_representation.items | ((SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
         'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
       * TYPEOF(it)) = 2) AND (it.name = 'target length')) )) = 1)) ))
         = 0)) )) = 0);
wr9 : (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
         QUERY ( pd <* USEDIN(SELF,
     'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
          | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
         'FEATURE_BASED_PROCESS_PLANNING.' +
         'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
         used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
         dtm_rep.used_representation.items | ((SIZEOF([
       'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
```

```
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'target length')) )) = 1)) ))
                = 0)) )) = 0);
    wr10: (NOT (SELF.description = 'rectangle')) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                dtm_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'target width')) )) = 1)) ))
                = 0)) )) = 0);
END_ENTITY; -- placed_datum_target_feature

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 | (
             'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'pocket depth occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | (((sar.description =
                'path feature component usage') AND (sar.name =
                'pocket depth')) AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((
                'FEATURE_BASED_PROCESS_PLANNING.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) |
       ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) )) = 1;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |  ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'boundary occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
                'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'closed boundary occurrence') AND
                 (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                 'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
                ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) |
                 ('FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'open boundary occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE' IN
                TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0);
wr10: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'bottom condition occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((
                'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN TYPEOF(
                sdr.relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF(sdr.
             related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
                = 0;
wr11: (NOT (SELF\characterized_object.description IN ['complex',
                'non-circular cutout','recess'])) OR (SIZEOF(
                QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'change in boundary occurrence')
                AND (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'taper usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((
                'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
                relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'enclosed boundary occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
                ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'bottom condition occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((
                'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN TYPEOF(
                sdr.relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'bottom condition occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((
                'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN TYPEOF(
                sdr.relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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) |
              (('FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | ((
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | (
                (('FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT'
                IN TYPEOF(csa)) AND (csa.name = 'uncut volume')) AND (SIZEOF(
                QUERY ( sar <* csa.component_relationships |
           (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar)) AND (SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.BOSS',
                'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION'] * TYPEOF(sar.
                related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1);
wr17: (NOT (SELF\characterized_object.description IN [
                'closed rectangular','open rectangular'])) OR (SIZEOF(
                QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'change in boundary occurrence')
                AND (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
            'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'taper usage') AND
            ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((
                'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
                relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF(fcr.
                related_shape_aspect.of_shape.definition))) AND (fcr.
                related_shape_aspect.description IN ['angle taper',
                'directed taper'])) )) >= 1)) )) <= 1)) )) = 0);
  END_ENTITY; -- pocket

  ENTITY pocket_bottom
    SUBTYPE OF (shape_aspect);
    WHERE
wr1 : 'FEATURE_BASED_PROCESS_PLANNING.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,
```



```
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.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,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             (('FEATURE_BASED_PROCESS_PLANNING.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,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                 'FEATURE_BASED_PROCESS_PLANNING.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,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                 used_representation)) )) = 1)) )) = 0);
wr7 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                 QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                 pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                  | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                 'FEATURE_BASED_PROCESS_PLANNING.' +
                 'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                 used_representation)) )) = 0)) )) = 0);
wr9 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                 QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```

```
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |  ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) =
                 0)) )) = 0);
wr10: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
                description = 'bottom condition occurrence') AND (
                'FEATURE_BASED_PROCESS_PLANNING.POCKET' IN TYPEOF(fcr.
                related_shape_aspect.of_shape.definition))) AND (
                'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM' IN TYPEOF(
                fcr.relating_shape_aspect))) )) >= 1;
wr11: (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'pocket bottom usage') AND
            ('FEATURE_BASED_PROCESS_PLANNING.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 (cartesian_point)
    SUBTYPE OF (geometric_representation_item);
  END_ENTITY; -- point

  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 product;
    id                 : identifier;
    name               : label;
    description        : OPTIONAL text;
    frame_of_reference : SET [1:?] OF product_context;
  END_ENTITY; -- product

  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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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_formation;
    id          : identifier;
    description : OPTIONAL text;
    of_product  : product;
  UNIQUE
    ur1 : id, of_product;
  END_ENTITY; -- product_definition_formation

  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([
    'FEATURE_BASED_PROCESS_PLANNING.CHARACTERIZED_PRODUCT_DEFINITION',
             'FEATURE_BASED_PROCESS_PLANNING.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],
                'FEATURE_BASED_PROCESS_PLANNING.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 profile_floor
    SUBTYPE OF (shape_aspect);
    WHERE
wr1 : 'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0);
wr4 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 0)) )) = 0);
wr6 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
```



```
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) =
                0)) )) = 0);
wr7 : SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | ((sar.description =
                'profile floor usage') AND
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((
                'FEATURE_BASED_PROCESS_PLANNING.OUTSIDE_PROFILE' IN TYPEOF(
                fcr.related_shape_aspect.of_shape.definition)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.PROFILE_FLOOR' IN TYPEOF(
                fcr.relating_shape_aspect))) )) >= 1;
wr8 : (NOT (SELF.description IN ['planar','complex'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
       ((('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.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,'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,'FEATURE_BASED_PROCESS_PLANNING.' +
               'DESCRIPTION_ATTRIBUTE.DESCRIBED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
               'NAME_ATTRIBUTE.NAMED_ITEM')) <= 1;
  END_ENTITY; -- property_definition_representation

  ENTITY protrusion
    SUBTYPE OF (feature_definition);
    WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
```



```
                sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
                QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'volume shape usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.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) |
               (('FEATURE_BASED_PROCESS_PLANNING.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 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRECISION_QUALIFIER' IN
                TYPEOF(temp)) )) < 2;
  END_ENTITY; -- qualified_representation_item

  ENTITY qualitative_uncertainty
    SUBTYPE OF (uncertainty_qualifier);
      uncertainty_value : text;
  END_ENTITY; -- qualitative_uncertainty

  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 ratio_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: 'FEATURE_BASED_PROCESS_PLANNING.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: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
               IN TYPEOF(SELF.of_shape.definition);
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION.DEFINITION') | (SIZEOF(QUERY ( pdr <*
               USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     (('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
                AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
      ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
                = 0;
  wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
      ((SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'length')) )) = 1)) )) = 0)) ))
                = 0;
  wr8: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'corner radius')) )) <= 1)) ))
                = 0)) )) = 0;
    END_ENTITY; -- rectangular_closed_profile

    ENTITY rectangular_pattern
```

```
    SUBTYPE OF (replicate_feature);
    WHERE
wr1 : SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') |
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar)) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT' IN TYPEOF(sdr
                .related_shape_aspect)) )) = 1) )) <= 5)) )) = 0;
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
           (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
              used_representation.name = 'row layout direction')) )) = 1)) ))
                = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          (('FEATURE_BASED_PROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'column layout direction')) )) =
                 1)) )) = 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```



```
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
            ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
            ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'row spacing')) )) = 1)) ))
                = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |  ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'column spacing')) )) = 1)) ))
                = 0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
```

```
              AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
  END_ENTITY; -- rectangular_pattern

  ENTITY referenced_modified_datum
    SUBTYPE OF (datum_reference);
      modifier : limit_condition;
  END_ENTITY; -- referenced_modified_datum

  ENTITY removal_volume
    SUBTYPE OF (feature_definition);
    WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
           ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'shape volume occurrence') AND (SIZEOF(
               QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'volume shape usage') AND ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (sdr.
              relating_shape_aspect.description = 'volume shape') )) = 1)) ))
               = 1)) )) = 0;
  END_ENTITY; -- removal_volume

  ENTITY replicate_feature
    SUPERTYPE OF (ONEOF (circular_pattern,rectangular_pattern,
        feature_pattern))
    SUBTYPE OF (feature_definition);
    WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr2: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') |
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) ) | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
               'FEATURE_BASED_PROCESS_PLANNING.INSTANCED_FEATURE'] *
```



```
                TYPEOF(fcr.related_shape_aspect)) >= 1) AND (fcr.name =
                'pattern basis')) )) = 1;
wr3: (SIZEOF(QUERY ( sar <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | (NOT
            ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) )) + SIZEOF(QUERY ( sar <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | (NOT
           ('FEATURE_BASED_PROCESS_PLANNING.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,'FEATURE_BASED_PROCESS_PLANNING.' +
                'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
      wr2: SIZEOF(USEDIN(SELF,'FEATURE_BASED_PROCESS_PLANNING.' +
                '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 representation_relationship;
      name        : label;
      description : OPTIONAL text;
      rep_1       : representation;
      rep_2       : representation;
```

```
  END_ENTITY; -- representation_relationship

  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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
               = 0;
wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
               SELF) |
         (('FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'outer edge shape occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
```



```
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'flat edge shape occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'rounded edge shape occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'sweep occurrence') AND (SIZEOF(
               QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.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) |
            (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
```

```
               IN TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'maximum feature limit')) )) >= 0;
  END_ENTITY; -- revolved_profile

  ENTITY rib_top
    SUBTYPE OF (feature_definition);
    WHERE
wr1: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
            ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'rib top condition occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'rib top usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP_FLOOR' IN TYPEOF(sdr
               .relating_shape_aspect)) AND (
               'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP' IN TYPEOF(sdr.
             related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
               = 0;
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
          (('FEATURE_BASED_PROCESS_PLANNING.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) |
            (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'maximum feature limit')) )) >= 0;
  END_ENTITY; -- rib_top

  ENTITY rib_top_floor
    SUBTYPE OF (shape_aspect);
    WHERE
wr1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
               IN TYPEOF(SELF.of_shape.definition);
wr2: SELF.description IN ['planar','complex'];
wr3: SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(SELF,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATING_SHAPE_ASPECT') | ((sar.description =
               'rib top usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.description
               = 'rib top condition occurrence') AND (
               'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP' IN TYPEOF(fcr.
               related_shape_aspect.of_shape.definition))) AND (
               'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP_FLOOR' IN TYPEOF(fcr
```



```
               .relating_shape_aspect))) )) >= 1;
wr4: (NOT (SELF.description = 'complex')) OR (SIZEOF(QUERY ( pd <*
               USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             (('FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'boundary occurrence') AND (SIZEOF(
               QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
               'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'diameter occurrence') AND (SIZEOF(
```

```
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE' IN
              TYPEOF(sdr.relating_shape_aspect)) AND (sdr.name =
              'diameter')) )) = 1)) )) = 1)) )) = 0;
wr2: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
             ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'hole depth occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'path feature component usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (((
              'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'bottom condition occurrence') AND (
              SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'hole bottom usage') AND
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
              'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM' IN TYPEOF(fcr.
              relating_shape_aspect)) AND (
              'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
            related_shape_aspect.of_shape.definition))) )) = 1)) )) = 1)) ))
              = 0;
wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'change in diameter occurrence') AND (
              SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'taper usage') AND
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
              IN TYPEOF(sar))) ) | ((
```



```
                'FEATURE_BASED_PROCESS_PLANNING.TAPER' IN TYPEOF(fcr.
                relating_shape_aspect)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.ROUND_HOLE' IN TYPEOF(fcr.
             related_shape_aspect.of_shape.definition))) )) = 1)) )) <= 1)) ))
                = 0;
wr5: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
                SELF) |
      ('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'partial circular boundary occurrence')
                AND (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'profile usage') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE' IN
                 TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (
                'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                 TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
                sa_occ.description = 'course of travel occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
                'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'path feature component usage') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | ((
                'FEATURE_BASED_PROCESS_PLANNING.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) |
```

```
            (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'maximum feature limit')) )) >= 0;
  END_ENTITY; -- rounded_end

  ENTITY rounded_u_profile
    SUBTYPE OF (shape_aspect);
    WHERE
wr1: 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
               IN TYPEOF(SELF.of_shape.definition);
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items |  ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
               = 0;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
```



```
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
             (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'profile limit')) )) <= 1)) )) =
                0;
    wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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 (('FEATURE_BASED_PROCESS_PLANNING.' +
                '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 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,'FEATURE_BASED_PROCESS_PLANNING.' +
                'ROLE_ASSOCIATION.ITEM_WITH_ROLE')) <= 1;
  END_ENTITY; -- security_classification_assignment

  ENTITY security_classification_level;
      name : label;
  END_ENTITY; -- security_classification_level

  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,'FEATURE_BASED_PROCESS_PLANNING.' +
             'ID_ATTRIBUTE.IDENTIFIED_ITEM')) <= 1;
END_ENTITY; -- shape_aspect

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,'FEATURE_BASED_PROCESS_PLANNING.' +
             '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: ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
             TYPEOF(SELF.definition)) OR (
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINITION' IN TYPEOF(
             SELF.definition.definition));
    wr2: 'FEATURE_BASED_PROCESS_PLANNING.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
         ('FEATURE_BASED_PROCESS_PLANNING.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([
            'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT',
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM']
               * TYPEOF(it)) = 1)) )) = 0;
  END_ENTITY; -- shape_representation_with_parameters

  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 slot
    SUBTYPE OF (feature_definition);
    WHERE
wr1: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
              sa_occ.description = 'swept shape occurrence') AND (SIZEOF(
              QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'profile usage') AND
             ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
              'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE'] *
              TYPEOF(sdr.relating_shape_aspect)) = 1) )) = 1)) )) = 1)) ))
              = 0;
wr3: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (
              'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
```

```
               sa_occ.description = 'course of travel occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'path feature component usage') AND ((sar.name =
               'course of travel') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar)))) ) | (
               'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT' IN
               TYPEOF(sdr.relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;

wr4: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
           ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
           IN TYPEOF(pd)) ) |
           (NOT((SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
           'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
           ((sa_occ.description = 'end condition occurrence') AND
           (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
           'FEATURE_BASED_PROCESS_PLANNING.'+
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           (((sar.description = 'slot end usage') AND
           (sar.name IN ['course of travel start','course of travel end']))
           AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
           IN TYPEOF(sar))) ) |
           ((('FEATURE_BASED_PROCESS_PLANNING.SLOT_END'
           IN TYPEOF(fcr.relating_shape_aspect)) AND
           (fcr.relating_shape_aspect.description
           IN ['open','radiused','flat','woodruff'])) AND
           ('FEATURE_BASED_PROCESS_PLANNING.SLOT'
           IN TYPEOF(fcr.related_shape_aspect.of_shape.definition)))
            )) = 1) ) )) = 2)
            OR
           (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
           'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
           ((sa_occ.description = 'end condition occurrence') AND
           (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
           'FEATURE_BASED_PROCESS_PLANNING.'+
           'SHAPE_ASPECT_RELATIONSHIP.RELATED_SHAPE_ASPECT') |
           (((sar.description = 'slot end usage') AND
           (sar.name IN ['course of travel start','course of travel end']))
           AND
           ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
           IN TYPEOF(sar))) ) |
           ((('FEATURE_BASED_PROCESS_PLANNING.SLOT_END'
           IN TYPEOF(fcr.relating_shape_aspect)) AND
           (fcr.relating_shape_aspect.description IN ['loop'])) AND
           ('FEATURE_BASED_PROCESS_PLANNING.SLOT'
           IN TYPEOF(fcr.related_shape_aspect.of_shape.definition)))
            )) = 1) ) )) = 1))) )) =0;

wr5: (NOT (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
        ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
         IN TYPEOF(pd)) ) |
```



```
        (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
        ((sa_occ.description = 'end condition occurrence') AND
        (SIZEOF(QUERY ( fcr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'+
        'RELATED_SHAPE_ASPECT') |
        (((sar.description = 'slot end usage') AND
        (sar.name IN ['course of travel start','course of travel end'])) AND
        ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
        IN TYPEOF(sar))) ) |
        ((('FEATURE_BASED_PROCESS_PLANNING.SLOT_END'
        IN TYPEOF(fcr.relating_shape_aspect)) AND
        (fcr.relating_shape_aspect.description IN ['loop'])) AND
        ('FEATURE_BASED_PROCESS_PLANNING.SLOT'
        IN TYPEOF(fcr.related_shape_aspect.of_shape.definition)))
        )) = 1)) )) = 1)) )) =0)
           OR
        (SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')|
        ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
         IN TYPEOF(pd)) ) |
        (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') |
        ((sa_occ.description = 'course of travel occurrence') AND
        (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
        'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'+
        'RELATED_SHAPE_ASPECT') |
        ((sar.description = 'path feature component usage') AND
        ((sar.name = 'course of travel') AND
        ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
        IN TYPEOF(sar)))) ) |
        (('FEATURE_BASED_PROCESS_PLANNING.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) | ((
           'FEATURE_BASED_PROCESS_PLANNING.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 : 'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
```

```
      ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) )) = 0)) )) = 0);
wr4 : (NOT (SELF.description IN ['flat','woodruff'])) OR (SIZEOF(
                QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
     ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) )) = 1)) )) = 0);
wr5 : (NOT (SELF.description IN ['flat'])) OR (SIZEOF(QUERY ( pd <*
                USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
   ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
                SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
    ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS'
                IN TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(
                QUERY ( it <* impl_rep.used_representation.items | ((
```



```
                'FEATURE_BASED_PROCESS_PLANNING.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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
      ('FEATURE_BASED_PROCESS_PLANNING.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,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATING_SHAPE_ASPECT') | (((sar.description =
                'slot end usage') AND (sar.name IN [
                'course of travel start','course of travel end'])) AND
          ('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (((fcr.related_shape_aspect.
                description = 'end condition occurrence') AND (
                'FEATURE_BASED_PROCESS_PLANNING.SLOT' IN TYPEOF(fcr.
                related_shape_aspect.of_shape.definition))) AND (
                'FEATURE_BASED_PROCESS_PLANNING.SLOT_END' IN TYPEOF(fcr.
                relating_shape_aspect))) )) >= 1;
  END_ENTITY; -- slot_end

  ENTITY solid_angle_measure_with_unit
    SUBTYPE OF (measure_with_unit);
    WHERE
      wr1: 'FEATURE_BASED_PROCESS_PLANNING.SOLID_ANGLE_UNIT' IN TYPEOF(
               SELF\measure_with_unit.unit_component);
  END_ENTITY; -- solid_angle_measure_with_unit

  ENTITY solid_angle_unit
    SUBTYPE OF (named_unit);
    WHERE
      wr1: ((((((SELF\named_unit.dimensions.length_exponent = 0) AND (SELF
                \named_unit.dimensions.mass_exponent = 0)) AND (SELF\
               named_unit.dimensions.time_exponent = 0)) AND (SELF\
               named_unit.dimensions.electric_current_exponent = 0)) AND (
               SELF\named_unit.dimensions.
               thermodynamic_temperature_exponent = 0)) AND (SELF\
               named_unit.dimensions.amount_of_substance_exponent = 0)) AND
                (SELF\named_unit.dimensions.luminous_intensity_exponent = 0);
  END_ENTITY; -- solid_angle_unit

  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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
```

```
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'radius')) )) = 1)) )) = 0)) ))
                = 0;
      wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.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 : 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
```



```
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
                AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
                = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'first radius')) )) <= 1)) ))
```

```
                = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'second radius')) )) <= 1)) ))
                = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
                = 0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
            'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
                = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'profile limit')) )) <= 1)) )) =
                 0;
wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
```



```
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'depth')) )) <= 1)) )) =
              0)) )) = 0;
  END_ENTITY; -- square_u_profile

  ENTITY standard_uncertainty
    SUPERTYPE OF (expanded_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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'course of travel occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'path feature component usage') AND
            ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | ((
               'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               sa_occ.description = 'removal boundary occurrence') AND (
               SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
               + 'RELATED_SHAPE_ASPECT') | ((sar.description =
               'profile usage') AND
            ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
               IN TYPEOF(sar))) ) | (
               'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE' IN TYPEOF(sdr.
```

```
               relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr4: SIZEOF(QUERY ( pdr <* get_property_definition_representations(
               SELF) |
             (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
               IN TYPEOF(pdr.used_representation)) AND (pdr.
               used_representation.name = 'maximum feature limit')) )) >= 0;
wr5: SIZEOF(QUERY ( pds <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | ((
               'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
                TYPEOF(pds)) AND (SIZEOF(QUERY ( csa <* USEDIN(pds,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE') | ((
               ('FEATURE_BASED_PROCESS_PLANNING.COMPOSITE_SHAPE_ASPECT' IN
               TYPEOF(csa)) AND (csa.name = 'uncut volume')) AND (SIZEOF(
               QUERY ( sar <* csa.component_relationships |
         (('FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_RELATIONSHIP'
               IN TYPEOF(sar)) AND (SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.BOSS',
               'FEATURE_BASED_PROCESS_PLANNING.PROTRUSION'] * TYPEOF(sar.
               related_shape_aspect)) = 1)) )) = 1)) )) <= 1)) )) = 1;
  END_ENTITY; -- step

  ENTITY straightness_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
      wr1: NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
               'GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE') IN TYPEOF(SELF));
  END_ENTITY; -- straightness_tolerance

  ENTITY surface
    SUPERTYPE OF (ONEOF (elementary_surface,swept_surface,bounded_surface))
    SUBTYPE OF (geometric_representation_item);
  END_ENTITY; -- surface

  ENTITY surface_curve
    SUBTYPE OF (curve);
      curve_3d              : curve;
      associated_geometry   : LIST [1:2] OF pcurve_or_surface;
      master_representation : preferred_surface_curve_representation;
    DERIVE
      basis_surface : SET [1:2] OF surface := get_basis_surface(SELF);
    WHERE
      wr1: curve_3d.dim = 3;
      wr2: ('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(
               associated_geometry[1])) OR (master_representation <>
               pcurve_s1);
      wr3: ('FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(
               associated_geometry[2])) OR (master_representation <>
               pcurve_s2);
      wr4: NOT ('FEATURE_BASED_PROCESS_PLANNING.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_profile_tolerance
    SUBTYPE OF (geometric_tolerance);
    WHERE
      wr1: (NOT (('FEATURE_BASED_PROCESS_PLANNING.' +
                '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 swept_surface
    SUPERTYPE OF (ONEOF (surface_of_linear_extrusion,surface_of_revolution))
    SUBTYPE OF (surface);
      swept_curve : curve;
  END_ENTITY; -- swept_surface

  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 taper
    SUBTYPE OF (shape_aspect);
    WHERE
wr1: 'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr4: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
```

```
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                 'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |  ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.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,
              'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.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 : 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
                 IN TYPEOF(SELF.of_shape.definition);
```



```
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
        'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((
               'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
               AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
               pd,'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
               impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
               'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
            * TYPEOF(it)) = 2) AND (it.name = 'width')) )) = 1)) )) = 0)) ))
```

```
                = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'depth')) )) = 1)) )) = 0)) ))
                = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'cross bar width')) )) =
                1)) )) = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'cross bar depth')) )) =
                1)) )) = 0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'first offset')) )) = 1)) ))
                = 0)) )) = 0;
wr11: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
```



```
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
             'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'second offset')) )) = 1)) ))
                = 0)) )) = 0;
wr12: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'first angle')) )) = 1)) ))
                = 0)) )) = 0;
wr13: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
             'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'second angle')) )) = 1)) ))
                = 0)) )) = 0;
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'radius')) )) <= 1)) )) =
                 0)) )) = 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
            (('FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
             * TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
                = 0)) )) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'minor diameter')) )) <=
                1)) )) = 0)) )) = 0;
wr4 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'pitch diameter')) )) <=
                1)) )) = 0)) )) = 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
```



```
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
            'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.RATIO_MEASURE_WITH_UNIT'] *
               TYPEOF(it)) = 2) AND (it.name = 'number of threads')) )) =
               1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
          (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
            IN TYPEOF(it)) AND (it.name = 'fit class')) )) = 1)) )) = 0)) ))
              = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (it.name = 'form')) )) = 1)) )) = 0)) ))
              = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
       ((('FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
```

```
        (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
              IN TYPEOF(it)) AND (it.name = 'qualifier')) )) <= 1)) )) =
              0)) )) = 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items |
         ((('FEATURE_BASED_PROCESS_PLANNING.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,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
              pd,'FEATURE_BASED_PROCESS_PLANNING.' +
              'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
              'FEATURE_BASED_PROCESS_PLANNING.' +
              'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
              used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
              impl_rep.used_representation.items | ((SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'crest')) )) <= 1)) )) =
              0)) )) = 0;
wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')|
              ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
               | ((sa_occ.description = 'partial area occurrence') AND (
              SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'applied area usage') AND
           ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (
              'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
              .relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr13: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
             ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
              IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
              pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
               | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
              + 'RELATED_SHAPE_ASPECT') | ((sar.description =
              'applied shape') AND
             ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
              IN TYPEOF(sar))) ) | (
              'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT' IN TYPEOF(sdr
              .relating_shape_aspect)) )) = 1) )) = 1)) )) = 0;
```



```
wr14: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
           (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name = 'fit class 2')) )) <= 1)) ))
                = 0)) )) = 0;
wr15: SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'nominal size')) )) <= 1)) ))
                = 0)) )) = 0;
wr16: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
               ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'thread runout') AND (SIZEOF(
                QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'thread runout usage') AND
              ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.THREAD_RUNOUT' IN TYPEOF(
                sdr.relating_shape_aspect)) )) <= 1)) )) = 1)) )) = 0;
  END_ENTITY; -- thread

  ENTITY thread_runout
    SUBTYPE OF (shape_aspect);
    WHERE
wr1: 'FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE' IN
               TYPEOF(SELF.of_shape);
wr2: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
               'FEATURE_BASED_PROCESS_PLANNING.' +
               'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
               used_representation)) )) = 1)) )) = 0;
wr3: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
               | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
```

```
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) AND (SIZEOF(QUERY ( srwp_i <* pdr.
                used_representation.items | (NOT (srwp_i.name IN [
             'length of runout','pitch or dimension','included or extra'])) ))
                > 0)) )) = 0) )) = 0;
wr5: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT'] *
              TYPEOF(it)) = 2) AND (it.name = 'length of runout')) )) = 1)) ))
                = 0)) )) <= 1;
wr6: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
         ((('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name = 'pitch or dimension')) AND (it
                .description IN ['pitch','pitch or dimension'])) )) = 1)) ))
                = 0)) )) = 0;
wr7: SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
          ((('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name = 'included or extra')) AND (it.
                description IN ['included','extra'])) )) = 1)) )) = 0)) )) =
                 0;
```



```
END_ENTITY; -- thread_runout

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 topological_representation_item
  SUPERTYPE OF (ONEOF (vertex,edge,face_bound,face,connected_face_set,
      loop ANDOR path))
  SUBTYPE OF (representation_item);
END_ENTITY; -- topological_representation_item

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(['FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION',
             'FEATURE_BASED_PROCESS_PLANNING.COMPOUND_FEATURE'] * TYPEOF(
             SELF.of_shape.definition)) = 1;
    wr2: SIZEOF(['FEATURE_BASED_PROCESS_PLANNING.CHAMFER',
             'FEATURE_BASED_PROCESS_PLANNING.EDGE_ROUND',
             'FEATURE_BASED_PROCESS_PLANNING.FILLET'] * TYPEOF(SELF)) = 1;
END_ENTITY; -- transition_feature
```

```
  ENTITY turned_knurl
    SUBTYPE OF (feature_definition);
    WHERE
wr1 : SELF\characterized_object.description IN ['diamond','diagonal',
                'straight'];
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
         'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'major diameter')) )) = 1)) ))
                = 0)) )) = 0;
wr5 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'nominal diameter')) )) =
                 1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
```

```
                'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.' +
                'DEFINITION') | (NOT (SIZEOF(QUERY ( impl_rep <*
                QUERY ( pdr <* USEDIN(pd,'FEATURE_BASED_PROCESS_PLANNING.'
                + 'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
               'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'tooth depth')) )) <= 1)) ))
                = 0)) )) = 0;
wr7 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.LENGTH_MEASURE_WITH_UNIT']
               * TYPEOF(it)) = 2) AND (it.name = 'root fillet')) )) <= 1)) ))
                = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
              'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.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,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
```

```
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
           (('FEATURE_BASED_PROCESS_PLANNING.DESCRIPTIVE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (it.name = 'helix hand')) )) = 1)) )) =
                0)) )) = 0);
wr11: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
               ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | ((sa_occ.description = 'partial area occurrence') AND (
                SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'applied area usage') AND ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_DEFINING_RELATIONSHIP') IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA' IN TYPEOF(sdr
                .relating_shape_aspect)) )) = 1)) )) = 1)) )) = 0;
wr12: SIZEOF(QUERY ( pds <* QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION') |
                 ('FEATURE_BASED_PROCESS_PLANNING.PRODUCT_DEFINITION_SHAPE'
                IN TYPEOF(pd)) ) | (NOT (SIZEOF(QUERY ( sa_occ <* USEDIN(
                pds,'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT.OF_SHAPE')
                 | (SIZEOF(QUERY ( sdr <* QUERY ( sar <* USEDIN(sa_occ,
               'FEATURE_BASED_PROCESS_PLANNING.SHAPE_ASPECT_RELATIONSHIP.'
                + 'RELATED_SHAPE_ASPECT') | ((sar.description =
                'applied shape') AND
               ('FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'
                IN TYPEOF(sar))) ) | (
                'FEATURE_BASED_PROCESS_PLANNING.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 (ONEOF (standard_uncertainty,qualitative_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 value_range
    SUBTYPE OF (compound_representation_item);
  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
             ('FEATURE_BASED_PROCESS_PLANNING.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 : 'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION'
                IN TYPEOF(SELF.of_shape.definition);
wr2 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
            'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) )) = 1)) )) = 0;
wr3 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                '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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | (((
                'FEATURE_BASED_PROCESS_PLANNING.' +
```

```
                '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,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((
                'FEATURE_BASED_PROCESS_PLANNING.PLACEMENT' IN TYPEOF(it))
                AND (it.name = 'orientation')) )) = 1)) )) = 0)) )) = 0;
wr6 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
          'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
             ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
              'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
              * TYPEOF(it)) = 2) AND (it.name = 'profile angle')) )) = 1)) ))
                = 0)) )) = 0;
wr8 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items | ((SIZEOF([
                'FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM',
```



```
             'FEATURE_BASED_PROCESS_PLANNING.PLANE_ANGLE_MEASURE_WITH_UNIT']
                * TYPEOF(it)) = 2) AND (it.name = 'tilt angle')) )) = 1)) ))
                = 0)) )) = 0;
wr9 : SIZEOF(QUERY ( pd <* USEDIN(SELF,
             'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( pdr <* USEDIN(pd,
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') |
              (('FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION'
                IN TYPEOF(pdr.used_representation)) AND (pdr.
                used_representation.name = 'profile limit')) )) <= 1)) )) =
                 0;
wr10: SIZEOF(QUERY ( pd <* USEDIN(SELF,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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,
           'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
                 | (NOT (SIZEOF(QUERY ( impl_rep <* QUERY ( pdr <* USEDIN(
                pd,'FEATURE_BASED_PROCESS_PLANNING.' +
                'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
                'FEATURE_BASED_PROCESS_PLANNING.' +
                'SHAPE_REPRESENTATION_WITH_PARAMETERS') IN TYPEOF(pdr.
                used_representation)) ) | (NOT (SIZEOF(QUERY ( it <*
                impl_rep.used_representation.items |
           ((('FEATURE_BASED_PROCESS_PLANNING.MEASURE_REPRESENTATION_ITEM'
                IN TYPEOF(it)) AND (
                'FEATURE_BASED_PROCESS_PLANNING.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 week_of_year_and_day_date
  SUBTYPE OF (date);
    week_component : week_in_year_number;
    day_component  : OPTIONAL day_in_week_number;
END_ENTITY; -- week_of_year_and_day_date

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 =
           'feature_based_process_planning')) )) = 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 chamfer_offset_requires_faces FOR (chamfer_offset,
           property_definition_representation);

WHERE
  wr1: SIZEOF(QUERY ( co <* chamfer_offset | (NOT (1 = SIZEOF(
           QUERY ( pdr <* property_definition_representation | (((pdr.
           definition.definition = co) AND (
           'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
           TYPEOF(pdr.used_representation))) AND (((pdr.definition.
           definition.description = 'first offset') AND (pdr.
           used_representation.name = 'first face shape')) OR ((pdr.
           definition.definition.description = 'second offset') AND (pdr.
```



```
          used_representation.name = 'second face shape')))) )))) )) = 0;

END_RULE; -- chamfer_offset_requires_faces

RULE chamfer_requires_faces FOR (chamfer,
          property_definition_representation);

WHERE
  wr1: SIZEOF(QUERY ( cf <* chamfer | (NOT (1 = SIZEOF(QUERY ( pdr <*
          property_definition_representation | (((pdr.definition.
          definition = cf) AND (
          'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
          TYPEOF(pdr.used_representation))) AND (pdr.used_representation
          .name = 'chamfer face')) )))) )) = 0;

END_RULE; -- chamfer_requires_faces

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_action_request_status FOR (
          action_request_status);

WHERE
  wr1: SIZEOF(QUERY ( arst <* action_request_status | (NOT (SIZEOF(
          USEDIN(arst,'')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_action_request_status

RULE dependent_instantiable_approval_status FOR (approval_status);

WHERE
  wr1: SIZEOF(QUERY ( ast <* approval_status | (NOT (SIZEOF(USEDIN(ast,
          '')) >= 1)) )) = 0;

END_RULE; -- dependent_instantiable_approval_status

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 edge_round_requires_faces FOR (edge_round,
           property_definition_representation);

WHERE
  wr1: 0 = SIZEOF(QUERY ( er <* edge_round | (NOT (1 = SIZEOF(
```



```
            QUERY ( pdr <* property_definition_representation | (((pdr.
            definition.definition = er) AND (
            'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation))) AND (pdr.used_representation
            .name = 'edge round face')) )))) ));
    wr2: SIZEOF(QUERY ( er <* edge_round | (NOT (1 = SIZEOF(
            QUERY ( pdr <* property_definition_representation | (((pdr.
            definition.definition = er) AND (
            'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation))) AND (pdr.used_representation
            .name = 'first face shape')) )))) )) = 0;
    wr3: SIZEOF(QUERY ( er <* edge_round | (NOT (1 = SIZEOF(
            QUERY ( pdr <* property_definition_representation | (((pdr.
            definition.definition = er) AND (
            'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION' IN
            TYPEOF(pdr.used_representation))) AND (pdr.used_representation
            .name = 'second face shape')) )))) )) = 0;

  END_RULE; -- edge_round_requires_faces

  RULE geometric_tolerance_subtype_exclusiveness FOR (geometric_tolerance);

  WHERE
    wr1: SIZEOF(QUERY ( gt <* geometric_tolerance | (NOT (SIZEOF(TYPEOF(gt)
            * ['FEATURE_BASED_PROCESS_PLANNING.ANGULARITY_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_RUNOUT_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.CONCENTRICITY_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.CYLINDRICITY_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.FLATNESS_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.LINE_PROFILE_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.PARALLELISM_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.PERPENDICULARITY_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.POSITION_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.ROUNDNESS_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.STRAIGHTNESS_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.SURFACE_PROFILE_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.SYMMETRY_TOLERANCE',
            'FEATURE_BASED_PROCESS_PLANNING.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,'FEATURE_BASED_PROCESS_PLANNING.' +
            '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_requires_externally_defined_representation FOR (
            product_definition,
            externally_defined_representation_with_parameters);

WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition | ((pdf.
            frame_of_reference.name = 'mating definition') AND (SIZEOF(
            QUERY ( pd <* USEDIN(pdf,'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION.DEFINITION') | (NOT (SIZEOF(
            QUERY ( dtm_rep <* QUERY ( pdr <* USEDIN(pd,
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'PROPERTY_DEFINITION_REPRESENTATION.DEFINITION') | ((
            'FEATURE_BASED_PROCESS_PLANNING.' +
            'EXTERNALLY_DEFINED_REPRESENTATION_WITH_PARAMETERS') IN
            TYPEOF(pdr.used_representation)) ) | (NOT (SIZEOF(dtm_rep.
            used_representation.items) >= 0)) )) = 0)) )) = 0)) )) = 0;

END_RULE; -- mating_definition_requires_externally_defined_representation

RULE non_instantiable_description_attribute FOR (description_attribute);

WHERE
  wr1: SIZEOF(QUERY ( da <* description_attribute | (NOT (SIZEOF(USEDIN(
            da,'')) = 1)) )) = 0;

END_RULE; -- non_instantiable_description_attribute

RULE non_instantiable_object_role FOR (object_role);

WHERE
  wr1: SIZEOF(QUERY ( da <* object_role | (NOT (SIZEOF(USEDIN(da,'')) =
            1)) )) = 0;

END_RULE; -- non_instantiable_object_role

RULE non_instantiable_role_association FOR (role_association);

WHERE
  wr1: SIZEOF(QUERY ( da <* role_association | (NOT (SIZEOF(USEDIN(da,''))
            = 1)) )) = 0;

END_RULE; -- non_instantiable_role_association

RULE part_requires_project_order FOR (product_definition_formation,
            feature_based_pp_action_assignment);

WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
            SIZEOF(QUERY ( fbppaa <* feature_based_pp_action_assignment |
```

```
          ((pdf IN fbppaa.items) AND (fbppaa.assigned_action.name =
          'project order')) )) = 1)) )) = 0;

END_RULE; -- part_requires_project_order

RULE part_to_approval FOR (product_definition_formation,
          feature_based_pp_approval_assignment);

WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
          SIZEOF(QUERY ( fbppa <* feature_based_pp_approval_assignment
           | (pdf IN fbppa.items) )) <= 1)) )) = 0;

END_RULE; -- part_to_approval

RULE product_definition_formation_requires_security_classification FOR (
          product_definition_formation,
          feature_based_pp_security_classification_assignment);

WHERE
  wr1: SIZEOF(QUERY ( pdf <* product_definition_formation | (NOT (
          SIZEOF(QUERY ( fbppsca <*
          feature_based_pp_security_classification_assignment | (pdf IN
          fbppsca.items) )) = 1)) )) = 0;

END_RULE; -- product_definition_formation_requires_security_classification

RULE product_requires_version FOR (product, product_definition_formation);

WHERE
  wr1: SIZEOF(QUERY ( prod <* product | (NOT (SIZEOF(QUERY ( pdf <*
          product_definition_formation | (prod :=: pdf.of_product) )) >=
           1)) )) = 0;

END_RULE; -- product_requires_version

RULE project_order_requires_approval FOR (directed_action,
          feature_based_pp_approval_assignment);

WHERE
  wr1: SIZEOF(QUERY ( po <* QUERY ( da <* directed_action | (da.name =
          'project order') ) | (NOT (SIZEOF(QUERY ( fbppapp <*
          feature_based_pp_approval_assignment | (po IN fbppapp.items) ))
          = 1)) )) = 0;

END_RULE; -- project_order_requires_approval

RULE project_order_tracking_relationships FOR (directed_action,
          action_relationship);

WHERE
  wr1: SIZEOF(QUERY ( da <* directed_action | ((da.name IN [
          'shop work order','resource acquisition order',
          'digital technical data package work order',
          'pedigree creation order']) AND (NOT (SIZEOF(QUERY ( ar <*
          action_relationship | ((da :=: ar.related_action) AND ((
          'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
```

```
            relating_action)) AND (ar.relating_action.name =
            'project order'))) )) = 1))) )) = 0;
    wr2: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'project order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.relating_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            related_action)) AND (ar.related_action.name =
            'shop work order'))) )) <= 1))) )) = 0;
    wr3: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'project order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.relating_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            related_action)) AND (ar.related_action.name =
            'resource acquisition order'))) )) <= 1))) )) = 0;
    wr4: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'project order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.relating_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            related_action)) AND (ar.related_action.name =
            'digital technical data package work order'))) )) <= 1))) )) =
             0;
    wr5: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'project order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.relating_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            related_action)) AND (ar.related_action.name =
            'pedigree creation order'))) )) <= 1))) )) = 0;
    wr6: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'customer order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.related_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            relating_action)) AND (ar.relating_action.name =
            'project order'))) )) >= 1))) )) = 0;
    wr7: SIZEOF(QUERY ( da <* directed_action | ((da.name =
            'project order') AND (NOT (SIZEOF(QUERY ( ar <*
            action_relationship | ((da :=: ar.relating_action) AND ((
            'FEATURE_BASED_PROCESS_PLANNING.DIRECTED_ACTION' IN TYPEOF(ar.
            related_action)) AND (ar.related_action.name =
            'customer order'))) )) <= 1))) )) = 0;

  END_RULE; -- project_order_tracking_relationships

  RULE representation_subtype_exclusiveness FOR (representation);

  WHERE
    wr1: SIZEOF(QUERY ( rep <* representation | (NOT (SIZEOF(TYPEOF(rep) *
             ['FEATURE_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION',

'FEATURE_BASED_PROCESS_PLANNING.DEFINITIONAL_REPRESENTATION','FEATURE_BASED_P
ROCESS_PLANNING.DIRECTION_SHAPE_REPRESENTATION',
            'FEATURE_BASED_PROCESS_PLANNING.FACE_SHAPE_REPRESENTATION',

'FEATURE_BASED_PROCESS_PLANNING.PLANAR_SHAPE_REPRESENTATION','FEATURE_BASED_P
ROCESS_PLANNING.LOCATION_SHAPE_REPRESENTATION',
            'FEATURE_BASED_PROCESS_PLANNING.PATH_SHAPE_REPRESENTATION'])
            <= 2)) )) = 0;
```



```
  END_RULE; -- representation_subtype_exclusiveness

  RULE restrict_approval_status FOR (approval_status);

  WHERE
    wr1: SIZEOF(QUERY ( ast <* approval_status | (NOT (ast.name IN [
             'approved','not yet approved','disapproved','withdrawn'])) ))
             = 0;

  END_RULE; -- restrict_approval_status

  RULE restrict_externally_defined_feature_definition FOR (
             externally_defined_feature_definition);

  WHERE
      wr1:((SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
         (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
         'FEATURE_BASED_PROCESS_PLANNING.APPLIED_DOCUMENT_REFERENCE.ITEMS')|
         ('FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE'
         IN TYPEOF(adr.assigned_document)))) >=1)))) =0) OR
         (SIZEOF(QUERY ( ex <* externally_defined_feature_definition |
         (NOT (SIZEOF(QUERY ( adr <* USEDIN(ex,
         'FEATURE_BASED_PROCESS_PLANNING'+
         '.APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT.ITEMS')|
         ('FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE'
         IN
TYPEOF(adr\document_usage_constraint_assignment.assigned_document_usage.sourc
e))))
         >=1)))) =0));
  END_RULE; -- restrict_externally_defined_feature_definition

  RULE restrict_security_classification_level FOR (
             security_classification_level);

  WHERE
    wr1: SIZEOF(QUERY ( scl <* security_classification_level | (NOT (scl.
             name IN ['unclassified','classified','proprietary',
             'confidential','secret','top_secret'])) )) = 0;

  END_RULE; -- restrict_security_classification_level

  RULE shape_aspect_relationship_subtype_exclusiveness FOR (
             shape_aspect_relationship);

  WHERE
    wr1: SIZEOF(QUERY ( sr <* shape_aspect_relationship | (NOT (SIZEOF(
             TYPEOF(sr) * [

'FEATURE_BASED_PROCESS_PLANNING.DIMENSIONAL_LOCATION','FEATURE_BASED_PROCESS_
PLANNING.GEOMETRIC_TOLERANCE_RELATIONSHIP','FEATURE_BASED_PROCESS_PLANNING.FE
ATURE_COMPONENT_RELATIONSHIP',
             'FEATURE_BASED_PROCESS_PLANNING.SHAPE_DEFINING_RELATIONSHIP'])
             <= 2)) )) = 0;

  END_RULE; -- shape_aspect_relationship_subtype_exclusiveness
```

```
RULE shape_aspect_subtype_exclusiveness FOR (shape_aspect);

WHERE
  wr1: SIZEOF(QUERY ( sr <* shape_aspect | (NOT (SIZEOF(TYPEOF(sr) * [
           'FEATURE_BASED_PROCESS_PLANNING.PATH_FEATURE_COMPONENT',
           'FEATURE_BASED_PROCESS_PLANNING.SLOT_END',
           'FEATURE_BASED_PROCESS_PLANNING.POCKET_BOTTOM',
           'FEATURE_BASED_PROCESS_PLANNING.PROFILE_FLOOR',
           'FEATURE_BASED_PROCESS_PLANNING.RIB_TOP_FLOOR',
           'FEATURE_BASED_PROCESS_PLANNING.BOSS_TOP',
           'FEATURE_BASED_PROCESS_PLANNING.HOLE_BOTTOM',
           'FEATURE_BASED_PROCESS_PLANNING.APPLIED_AREA',
           'FEATURE_BASED_PROCESS_PLANNING.TAPER',
           'FEATURE_BASED_PROCESS_PLANNING.CHAMFER_OFFSET',
           'FEATURE_BASED_PROCESS_PLANNING.CIRCULAR_CLOSED_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.NGON_CLOSED_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.CLOSED_PATH_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.SQUARE_U_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.LINEAR_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.TEE_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.VEE_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.RECTANGULAR_CLOSED_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.PARTIAL_CIRCULAR_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.ROUNDED_U_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.OPEN_PATH_PROFILE',
           'FEATURE_BASED_PROCESS_PLANNING.INSTANCED_FEATURE',
           'FEATURE_BASED_PROCESS_PLANNING.REPLICATE_FEATURE',
           'FEATURE_BASED_PROCESS_PLANNING.TRANSITION_FEATURE',
           'FEATURE_BASED_PROCESS_PLANNING.DATUM',
           'FEATURE_BASED_PROCESS_PLANNING.DATUM_FEATURE',
           'FEATURE_BASED_PROCESS_PLANNING.DATUM_TARGET']) <= 2)) )) = 0;

END_RULE; -- shape_aspect_subtype_exclusiveness

RULE shape_representation_subtype_exclusiveness FOR (
           shape_representation);

WHERE
  wr1: SIZEOF(QUERY ( sr <* shape_representation | (NOT (SIZEOF(TYPEOF(
           sr)                                                           *
['FEATURE_BASED_PROCESS_PLANNING.ADVANCED_BREP_SHAPE_REPRESENTATION','FEATURE
_BASED_PROCESS_PLANNING.SHAPE_REPRESENTATION_WITH_PARAMETERS','FEATURE_BASED_
PROCESS_PLANNING.SHAPE_DIMENSION_REPRESENTATION'])
           <= 2)) )) = 0;

END_RULE; -- shape_representation_subtype_exclusiveness

RULE subtype_mandatory_characterized_object FOR (characterized_object);

WHERE
  wr1: SIZEOF(QUERY ( csa <* characterized_object | ((NOT (SIZEOF([
           'FEATURE_BASED_PROCESS_PLANNING.DOCUMENT_FILE',
           'FEATURE_BASED_PROCESS_PLANNING.FEATURE_DEFINITION',
           'FEATURE_BASED_PROCESS_PLANNING.FEATURE_COMPONENT_DEFINITION',
           'FEATURE_BASED_PROCESS_PLANNING.ORDERED_PART'] * TYPEOF(csa))
```



```
            =     1))     AND     (NOT     (SIZEOF(QUERY     (     pd     <*
USEDIN(csa,'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITION.DEFINITION')
             | ('FEATURE_BASED_PROCESS_PLANNING.MATERIAL_PROPERTY' IN
            TYPEOF(pd)) )) = 1))) )) = 0;

  END_RULE; -- subtype_mandatory_characterized_object

  RULE transition_feature_life_cycle FOR (transition_feature);

  WHERE
    wr1: SIZEOF(QUERY ( tf <* transition_feature | (NOT (tf.of_shape.
             definition.frame_of_reference.life_cycle_stage =
             'manufacturing planning')) )) = 0;

  END_RULE; -- transition_feature_life_cycle

  RULE transition_feature_on_part_boundary FOR (transition_feature);

  WHERE
    wr1: SIZEOF(QUERY ( tf <* transition_feature | (NOT tf.
             product_definitional) )) = 0;

  END_RULE; -- transition_feature_on_part_boundary

  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 | (
        'FEATURE_BASED_PROCESS_PLANNING.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],'')) | (
            'FEATURE_BASED_PROCESS_PLANNING.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_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,('FEATURE_BASED_PROCESS_PLANNING.' +
        '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 associated_surface(
               arg: pcurve_or_surface
      ): surface;

    LOCAL
      surf : surface;
    END_LOCAL;
    IF 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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_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 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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.CARTESIAN_POINT' IN TYPEOF(item)
      THEN
    dim := SIZEOF(item\cartesian_point.coordinates);
    RETURN(dim);
  END_IF;
  IF 'FEATURE_BASED_PROCESS_PLANNING.DIRECTION' IN TYPEOF(item) THEN
    dim := SIZEOF(item\direction.direction_ratios);
    RETURN(dim);
  END_IF;
  IF 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 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 'FEATURE_BASED_PROCESS_PLANNING.PCURVE' IN TYPEOF(c) THEN
    surfs := [c\pcurve.basis_surface];
  ELSE
    IF 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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,(
                        'FEATURE_BASED_PROCESS_PLANNING.' +
                        '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,(
                 'FEATURE_BASED_PROCESS_PLANNING.' + '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,(
                   'FEATURE_BASED_PROCESS_PLANNING.' + '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,
        'FEATURE_BASED_PROCESS_PLANNING.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],'FEATURE_BASED_PROCESS_PLANNING.PROPERTY_DEFINITI
ON_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,(
                  'FEATURE_BASED_PROCESS_PLANNING.' + '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,
        'FEATURE_BASED_PROCESS_PLANNING.REPRESENTATION.ITEMS') * cntxt.
        representations_in_context) > 0 THEN
      RETURN(TRUE);
    ELSE
      y := QUERY ( z <* USEDIN(item,'') | (
          'FEATURE_BASED_PROCESS_PLANNING.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 := 'FEATURE_BASED_PROCESS_PLANNING.POLY_LOOP' IN
      TYPEOF(l[1]);
  REPEAT i := 2 TO SIZEOF(l) BY 1;
    IF ('FEATURE_BASED_PROCESS_PLANNING.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 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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.OPEN_SHELL' IN TYPEOF(a_shell)
       THEN
    RETURN(open_shell_reversed(a_shell));
  ELSE
    IF 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.EDGE' IN TYPEOF(an_item) THEN
      RETURN(edge_reversed(an_item));
    END_IF;
    IF 'FEATURE_BASED_PROCESS_PLANNING.PATH' IN TYPEOF(an_item) THEN
      RETURN(path_reversed(an_item));
    END_IF;
    IF 'FEATURE_BASED_PROCESS_PLANNING.FACE_BOUND' IN TYPEOF(an_item)
         THEN
      RETURN(face_bound_reversed(an_item));
    END_IF;
    IF 'FEATURE_BASED_PROCESS_PLANNING.FACE' IN TYPEOF(an_item) THEN
      RETURN(face_reversed(an_item));
    END_IF;
    IF 'FEATURE_BASED_PROCESS_PLANNING.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 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,'')) | ((
        'FEATURE_BASED_PROCESS_PLANNING.REPRESENTATION_ITEM' IN TYPEOF(z))
        OR ('FEATURE_BASED_PROCESS_PLANNING.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,
        'FEATURE_BASED_PROCESS_PLANNING.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],
            'FEATURE_BASED_PROCESS_PLANNING.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_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_units(
             m: measure_with_unit
      ): BOOLEAN;
    IF 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 'FEATURE_BASED_PROCESS_PLANNING.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 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 'FEATURE_BASED_PROCESS_PLANNING.VECTOR' IN TYPEOF(arg1) THEN
            mag1 := arg1.magnitude;
            vec1 := arg1.orientation;
          ELSE
            mag1 := 1;
            vec1 := arg1;
          END_IF;
          IF 'FEATURE_BASED_PROCESS_PLANNING.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; -- feature_based_process_planning
```

Annex B
(normative)
AIM short names

Table B.1 provides the short names of entities specified in the AIM of this part of ISO 10303. Requirements on the use of the short names are found in the implementation methods included in ISO 10303.

Table B.1 — AIM short names of entities

AIM Element	Short name
ACTION	ACTION
ACTION_ASSIGNMENT	ACTASS
ACTION_DIRECTIVE	ACTDRC
ACTION_METHOD	ACTMTH
ACTION_RELATIONSHIP	ACTRLT
ACTION_REQUEST_ASSIGNMENT	ACRQAS
ACTION_REQUEST_SOLUTION	ACRQSL
ACTION_REQUEST_STATUS	ACRQST
ACTION_STATUS	ACTSTT
ADDRESS	ADDRSS
ADVANCED_BREP_SHAPE_REPRESENTATION	ABSR
ADVANCED_FACE	ADVFC
ANGULAR_LOCATION	ANGLCT
ANGULAR_SIZE	ANGSZ
ANGULARITY_TOLERANCE	ANGTLR
APPLICATION_CONTEXT	APPCNT
APPLICATION_CONTEXT_ELEMENT	APCNEL
APPLICATION_PROTOCOL_DEFINITION	APPRDF
APPLIED_AREA	APPAR
APPLIED_IDENTIFICATION_ASSIGNMENT	APIDAS
APPLIED_CLASSIFICATION_ASSIGNMENT	APCLAS
APPLIED_DOCUMENT_REFERENCE	APDCRF

Table B.1 – AIM short names of entities (continued)

APPLIED_DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	ADUCA
APPLIED_EXTERNAL_IDENTIFICATION_ASSIGNMENT	AEIA
APPLIED_GROUP_ASSIGNMENT	APGRAS
APPROVAL	APPRVL
APPROVAL_ASSIGNMENT	APPASS
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
BOUNDED_CURVE	BNDCRV
BOUNDED_SURFACE	BNDSRF
BREP_WITH_VOIDS	BRWTVD
CALENDAR_DATE	CLNDT
CARTESIAN_POINT	CRTPNT
CARTESIAN_TRANSFORMATION_OPERATOR	CRTROP
CARTESIAN_TRANSFORMATION_OPERATOR_3D	CTO3

Table B.1 – AIM short names of entities (continued)

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
COMPOSITE_CURVE	CMPCRV
COMPOSITE_CURVE_SEGMENT	CMCRSG
COMPOSITE_HOLE	CMPHL
COMPOSITE_SHAPE_ASPECT	CMSHAS
COMPOUND_FEATURE	CMPFTR
CONCENTRICITY_TOLERANCE	CNCTLR
CONIC	CONIC
CONICAL_SURFACE	CNCSRF
CONNECTED_FACE_SET	CNFCST
CONTEXT_DEPENDENT_UNIT	CNDPUN
CONVERSION_BASED_UNIT	CNBSUN
CURVE	CURVE
CYLINDRICAL_SHAPE_REPRESENTATION	CYSHRP
CYLINDRICAL_SURFACE	CYLSRF
CYLINDRICITY_TOLERANCE	CYLTLR
DATA_ENVIRONMENT	DTENV
DATE	DATE

Table B.1 – AIM short names of entities (continued)

DATE_ASSIGNMENT	DTASS
DATE_ROLE	DTRL
DATUM	DATUM
DATUM_FEATURE	DTMFTR
DATUM_REFERENCE	DTMRFR
DATUM_TARGET	DTMTRG
DEFINITIONAL_REPRESENTATION	DFNRPR
DEGENERATE_TOROIDAL_SURFACE	DGTRSR
DERIVED_UNIT	DRVUNT
DERIVED_UNIT_ELEMENT	DRUNEL
DESCRIPTION_ATTRIBUTE	DSCATT
DESCRIPTIVE_REPRESENTATION_ITEM	DSRPIT
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_REFERENCE	DCMRFR
DOCUMENT_RELATIONSHIP	DCMRLT
DOCUMENT_REPRESENTATION_TYPE	DCRPTY
DOCUMENT_TYPE	DCMTYP
DOCUMENT_USAGE_CONSTRAINT	DCUSCN

Table B.1 – AIM short names of entities (continued)

DOCUMENT_USAGE_CONSTRAINT_ASSIGNMENT	DUCA
DOCUMENT_USAGE_ROLE	DCUSRL
DOCUMENT_WITH_CLASS	DCWTCL
EDGE	EDGE
EDGE_CURVE	EDGCRV
EDGE_LOOP	EDGLP
EDGE_ROUND	EDGRND
ELEMENTARY_SURFACE	ELMSRF
ELLIPSE	ELLPS
EXECUTED_ACTION	EXCACT
EXPANDED_UNCERTAINTY	EXPUNC
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
FACE	FACE
FACE_BOUND	FCBND
FACE_OUTER_BOUND	FCOTBN
FACE_SHAPE_REPRESENTATION	FCSHRP
FACE_SHAPE_REPRESENTATION_RELATIONSHIP	FSRR
FACE_SURFACE	FCSRF
FEATURE_BASED_PP_ACTION_ASSIGNMENT	FBPAA
FEATURE_BASED_PP_ACTION_REQUEST_ASSIGNMENT	FBPARA
FEATURE_BASED_PP_APPROVAL_ASSIGNMENT	FBP0

Table B.1 – AIM short names of entities (continued)

FEATURE_BASED_PP_DATE_ASSIGNMENT	FBPDA
FEATURE_BASED_PP_ORGANIZATION_ASSIGNMENT	FBPOA
FEATURE_BASED_PP_PERSON_AND_ORGANIZATION_ASSIGNMENT	FBPPAO
FEATURE_BASED_PP_SECURITY_CLASSIFICATION_ASSIGNMENT	FBPSCA
FEATURE_COMPONENT_DEFINITION	FTCMDF
FEATURE_COMPONENT_RELATIONSHIP	FTCMRL
FEATURE_DEFINITION	FTRDFN
FEATURE_PATTERN	FTRPTT
FILLET	FILLET
FLAT_FACE	FLTFC
FLATNESS_TOLERANCE	FLTTLR
FOUNDED_ITEM	FNDITM
FUNCTIONALLY_DEFINED_TRANSFORMATION	FNDFTR
GEAR	GEAR
GENERAL_PROPERTY	GNRPRP
GENERAL_PROPERTY_ASSOCIATION	GNPRAS
GEOMETRIC_REPRESENTATION_CONTEXT	GMRPCN
GEOMETRIC_REPRESENTATION_ITEM	GMRPIT
GEOMETRIC_TOLERANCE	GMTTLR
GEOMETRIC_TOLERANCE_RELATIONSHIP	GMTLRL
GEOMETRIC_TOLERANCE_WITH_DATUM_REFERENCE	GTWDR
GEOMETRIC_TOLERANCE_WITH_DEFINED_UNIT	GTWDU
GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT	GC
GLOBAL_UNIT_ASSIGNED_CONTEXT	GUAC
GROUP	GROUP
GROUP_ASSIGNMENT	GRPASS
GROUP_RELATIONSHIP	GRPRLT
HOLE_BOTTOM	HLBTT
HYPERBOLA	HYPRBL

Table B.1 – AIM short names of entities (continued)

ID_ATTRIBUTE	IDATT
IDENTIFICATION_ASSIGNMENT	IDNASS
IDENTIFICATION_ROLE	IDNRL
INSTANCED_FEATURE	INSFTR
KNOWN_SOURCE	KNWSRC
LENGTH_MEASURE_WITH_UNIT	LMWU
LENGTH_UNIT	LNGUNT
LIBRARY_CLASS_VERSION_ASSIGNMENT	LCVA
LIBRARY_PROPERTY_VERSION_ASSIGNMENT	LPVA
LIMITS_AND_FITS	LMANFT
LINE	LINE
LINE_PROFILE_TOLERANCE	LNP0
LINEAR_PROFILE	LNRPRF
LOCATION_SHAPE_REPRESENTATION	LCSHRP
LOOP	LOOP
MAKE_FROM_USAGE_OPTION	MFUO
MANIFOLD_SOLID_BREP	MNSLBR
MAPPED_ITEM	MPPITM
MARKING	MRKNG
MATERIAL_DESIGNATION	MTRDSG
MATERIAL_PROPERTY	MTRPRP
MATERIAL_PROPERTY_REPRESENTATION	MTPRRP
MEASURE_QUALIFICATION	MSRQLF
MEASURE_REPRESENTATION_ITEM	MSRPIT
MEASURE_WITH_UNIT	MSWTUN
MODIFIED_GEOMETRIC_TOLERANCE	MDGMTL
MODIFIED_PATTERN	MDFPTT
NAME_ATTRIBUTE	NMATT
NAMED_UNIT	NMDUNT

Table B.1 – AIM short names of entities (continued)

NEXT_ASSEMBLY_USAGE_OCCURRENCE	NAUO
NGON_CLOSED_PROFILE	NGCLPR
NGON_SHAPE_REPRESENTATION	NGSHRP
OBJECT_ROLE	OBJRL
OPEN_PATH_PROFILE	OPPTPR
OPEN_SHELL	OPNSHL
ORDERED_PART	ORDPRT
ORDINAL_DATE	ORDDT
ORGANIZATION	ORGNZT
ORGANIZATION_ASSIGNMENT	ORGASS
ORGANIZATION_ROLE	ORGRL
ORGANIZATIONAL_ADDRESS	ORGADD
ORIENTED_CLOSED_SHELL	ORCLSH
ORIENTED_EDGE	ORNEDG
ORIENTED_FACE	ORNFC
ORIENTED_OPEN_SHELL	OROPSH
ORIENTED_PATH	ORNPTH
OUTER_ROUND	OTRRND
OUTSIDE_PROFILE	OTSPRF
PARABOLA	PRBL
PARALLELISM_TOLERANCE	PRLTLR
PARAMETRIC_REPRESENTATION_CONTEXT	PRRPCN
PARTIAL_CIRCULAR_PROFILE	PRCRPR
PATH	PATH
PATH_FEATURE_COMPONENT	PTFTCM
PATH_SHAPE_REPRESENTATION	PTSHRP
PATTERN_OFFSET_MEMBERSHIP	PTOFMM
PATTERN_OMIT_MEMBERSHIP	PTOMMM
PCURVE	PCURVE

Table B.1 – AIM short names of entities (continued)

PERPENDICULARITY_TOLERANCE	PRPTLR
PERSON	PERSON
PERSON_AND_ORGANIZATION	PRANOR
PERSON_AND_ORGANIZATION_ASSIGNMENT	PAOA
PERSON_AND_ORGANIZATION_ROLE	PAOR
PERSONAL_ADDRESS	PRSADD
PLACED_DATUM_TARGET_FEATURE	PDT0
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
POLYLINE	PLYLN
POSITION_TOLERANCE	PSTTLR
PRE_DEFINED_ITEM	PRDFIT
PRECISION_QUALIFIER	PRCQLF
PRODUCT	PRDCT
PRODUCT_CONTEXT	PRDCNT
PRODUCT_DEFINITION	PRDDFN
PRODUCT_DEFINITION_CONTEXT	PRDFCN
PRODUCT_DEFINITION_FORMATION	PRDFFR
PRODUCT_DEFINITION_RELATIONSHIP	PRDFRL
PRODUCT_DEFINITION_SHAPE	PRDFSH
PRODUCT_DEFINITION_USAGE	PRDFUS
PRODUCT_DEFINITION_WITH_ASSOCIATED_DOCUMENTS	PDWAD

Table B.1 – AIM short names of entities (continued)

PROFILE_FLOOR	PRFFLR
PROJECTED_ZONE_DEFINITION	PRZNDF
PROPERTY_DEFINITION	PRPDFN
PROPERTY_DEFINITION_RELATIONSHIP	PRDFR
PROPERTY_DEFINITION_REPRESENTATION	PRDFRP
PROTRUSION	PRTRSN
QUALIFIED_REPRESENTATION_ITEM	QLRPIT
QUALITATIVE_UNCERTAINTY	QLTUNC
QUASI_UNIFORM_CURVE	QSUNCR
QUASI_UNIFORM_SURFACE	QSUNSR
RATIO_MEASURE_WITH_UNIT	RMWU
RATIO_UNIT	RTUNT
RATIONAL_B_SPLINE_CURVE	RBSC
RATIONAL_B_SPLINE_SURFACE	RBSS
RECTANGULAR_CLOSED_PROFILE	RCCLPR
RECTANGULAR_PATTERN	RCTPTT
REFERENCED_MODIFIED_DATUM	RFMDDT
REMOVAL_VOLUME	RMVVLM
REPLICATE_FEATURE	RPLFTR
REPRESENTATION	RPRSNT
REPRESENTATION_CONTEXT	RPRCNT
REPRESENTATION_ITEM	RPRITM
REPRESENTATION_MAP	RPRMP
REPRESENTATION_RELATIONSHIP	RPRRLT
REVOLVED_PROFILE	RVLPRF
RIB_TOP	RBTP
RIB_TOP_FLOOR	RBTPFL
ROLE_ASSOCIATION	RLASS
ROUND_HOLE	RNDHL

Table B.1 – AIM short names of entities (continued)

ROUNDED_END	RNDEND
ROUNDED_U_PROFILE	RNUPR
ROUNDNESS_TOLERANCE	RNDTLR
RUNOUT_ZONE_DEFINITION	RNZNDF
RUNOUT_ZONE_ORIENTATION	RNZNOR
SECURITY_CLASSIFICATION	SCRCLS
SECURITY_CLASSIFICATION_ASSIGNMENT	SCCLAS
SECURITY_CLASSIFICATION_LEVEL	SCCLLV
SHAPE_ASPECT	SHPASP
SHAPE_ASPECT_RELATIONSHIP	SHASRL
SHAPE_DEFINING_RELATIONSHIP	SHDFRL
SHAPE_DEFINITION_REPRESENTATION	SHDFRP
SHAPE_DIMENSION_REPRESENTATION	SHDMRP
SHAPE_REPRESENTATION	SHPRPR
SHAPE_REPRESENTATION_WITH_PARAMETERS	SRWP
SI_UNIT	SUNT
SLOT	SLOT
SLOT_END	SLTEND
SOLID_ANGLE_MEASURE_WITH_UNIT	SAMWU
SOLID_ANGLE_UNIT	SLANUN
SOLID_MODEL	SLDMDL
SPHERICAL_CAP	SPHCP
SPHERICAL_SURFACE	SPHSRF
SQUARE_U_PROFILE	SQUPR
STANDARD_UNCERTAINTY	STNUNC
STEP	STEP
STRAIGHTNESS_TOLERANCE	STRTLR
SURFACE	SRFC
SURFACE_CURVE	SRFCRV

Table B.1 – AIM short names of entities (continued)

SURFACE_OF_LINEAR_EXTRUSION	SL
SURFACE_OF_REVOLUTION	SROFRV
SURFACE_PROFILE_TOLERANCE	SRPRTL
SWEPT_SURFACE	SWPSRF
SYMMETRY_TOLERANCE	SYMTLR
TAPER	TAPER
TEE_PROFILE	TPRF
THREAD	THREAD
THREAD_RUNOUT	THRRNT
TOLERANCE_VALUE	TLRVL
TOLERANCE_ZONE	TLRZN
TOLERANCE_ZONE_DEFINITION	TLZNDF
TOLERANCE_ZONE_FORM	TLZNFR
TOPOLOGICAL_REPRESENTATION_ITEM	TPRPIT
TOROIDAL_SURFACE	TRDSRF
TOTAL_RUNOUT_TOLERANCE	TTRNTL
TRANSITION_FEATURE	TRNFTR
TURNED_KNURL	TRNKNR
TYPE_QUALIFIER	TYPQLF
UNCERTAINTY_MEASURE_WITH_UNIT	UMWU
UNCERTAINTY_QUALIFIER	UNCQLF
UNIFORM_CURVE	UNFCRV
UNIFORM_SURFACE	UNFSRF
VALUE_RANGE	VLRNG
VALUE_REPRESENTATION_ITEM	VLRPIT
VECTOR	VECTOR
VEE_PROFILE	VPRF
VERSIONED_ACTION_REQUEST	VRACRQ
VERTEX	VERTEX

Table B.1 – AIM short names of entities (continued)

VERTEX_LOOP	VRTLP
VERTEX_POINT	VRTPNT
WEEK_OF_YEAR_AND_DAY_DATE	WOYADD

Annex C
(normative)
Implementation method specific requirements

The implementation method defines what type of exchange behavior is required with respect to this part of ISO 10303. Conformance to this part of ISO 10303 shall be realized in an exchange structure. The file format shall be encoded according to the syntax and EXPRESS language mapping defined in ISO 10303-21 and annotated listing defined in Annex A of this part of ISO 10303. The header of the exchange structure shall identify use of this part of ISO 10303 by the schema name 'feature_based_process_-planning'.

Annex D
(normative)
Protocol Implementation Conformance Statement (PICS) proforma

This clause list the optional elements of this part of ISO 10303. An implementation may chose 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 filled out by the implementor and may be used in preparation for conformance testing by a testing laboratory. The completed PICS proforma is referred to as a PICS.

The information contained in the PICS is used to configure as appropriate executable test suite for use by the client.

A single conformance class is identified in this part of ISO 10303. A conforming implementation shall support this one conformance class. This class is detailed in Clause 6 of ISO 10303-224.

Question:

1. Please provide an identifier for the product or system for which conformance is claimed:

    Product name and current version number:______________________________

2. Please indicate the implementation method chosen:

    ISO 10303-21 Exchange Structure -- preprocessor
    Preprocessor name and current version number:___________________________

    ISO 10303-21 Exchange Structure -- postprocessor
    Postprocessor name and current version number:___________________________

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(224) version (3) }

is assigned to this part of ISO 10303. The meaning of this value is defined in ISO/IEC 8824-1, and described in ISO 10303-1.

E.2 Schema identification

To provide for unambiguous identification of the schema specification given in this application protocol feature-based-process-planning-schema in the open information system, the object identifier are assigned as follows:

{ iso standard 10303 part(224) version(3) schema(1) feature-based-process-planning-schema-lf(1) }

is assigned to the feature_based_process_planning long form schema (see Annex A).

{ iso standard 10303 part(224) version(3) schema(1) feature-based-process-planning-schema(2) }

is assigned to the feature_based_process_planning schema short form schema (see 5.2).

The meaning of this value is defined in ISO/IEC 8824-1, and is described in ISO 10303-1.

Annex F
(informative)
Application activity model

The application activity model (AAM) is provided to aid in the understanding the scope and information requirements defined in this application protocol. The model is presented as a set of activity figures that contain the activity diagrams and a set of definitions of the activities and their data.

The AAM covers activities which go beyond the subject of this application protocol. The diagrams use a modified IDEF0 notation [1]. 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 an enterprise for manufacturing a part. There are several activity diagrams that have all activities out of scope but they are important in illustrating how the manufacture of a part process was developed and how the in-scope requirements were derived.

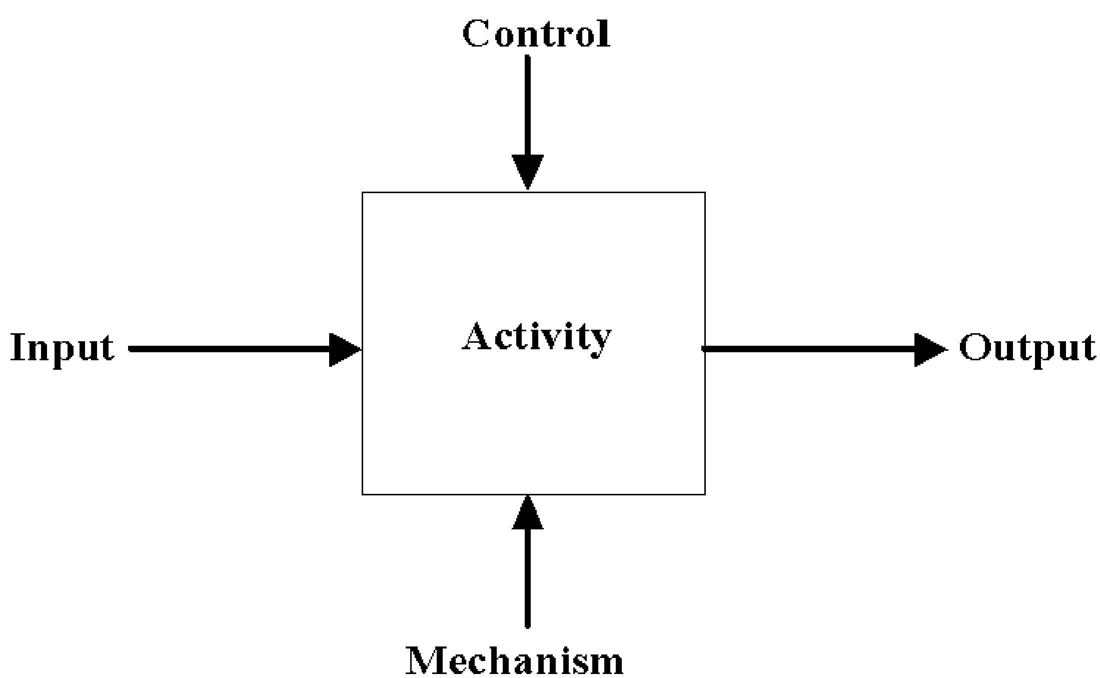


Figure F.1 — IDEF0 basic notation

F.1 Application activity model definitions

The following terms are used in the application activity model. Terms marked with an asterisk are outside the scope of this application protocol.

The definitions in this annex do not supersede the definitions given in the main body of the text.

F.1.1 Application activity model activity definitions

The following terms are used to define the activities of the AAM diagrams.

F.1.1.1 Capture Digital Product Data Definition (A2): Executes the methods available, for transforming part technical data into ISO 10303 product data.

F.1.1.2 Capture Notes (A2141): Compiles non-graphical information contained in the part information packet and represents it in the digital part model.

F.1.1.3 Capture Part Data Definition (A21): Examines the part information packet, determines the required information for a complete part definition, and creates digital part data definition.

F.1.1.4 Capture Part Management Data (A211): Collects product life cycle information associated with the part design from the part information packet.

F.1.1.5 Capture Part Shape Data (A2142): Interprets the graphical shape representation of the part from the part information packet, and creates an equivalent shape representation in the digital part model.

F.1.1.6 Capture Part Tolerance (A2143): Defines dimensional and geometric tolerance information contained in the part information packet in the digital part model.

F.1.1.7 Compile Part Pedigree Data (A42)*: Tracks and compiles required historical information pertaining to a part throughout the entire manufacturing process as defined by a customer.

F.1.1.8 Control Workstations and Coordinate Transportation (A41)*: Provides all of the information processing required to support the execution of production tasks and to support the transport of items.

F.1.1.9 Create Digital Product Data Definition (A25): Creates digital product data definition for a specific part based on the part information packet.

F.1.1.10 Create Manufacturing Bill of Materials (A314)*: Generates a list of raw stock and indirect material items that must be available prior to manufacturing the part.

F.1.1.11 Create Part Model (A214): Develops a digital model of a part from the information contained in the part information packet.

F.1.1.12 Create Shop Work Order (A32)*: Prepares a shop work agenda and associated information required to route the part through the shop floor for manufacture.

F.1.1.13 Define Cutting Tools (A3111)*: Selects the machine tools required to perform machining operations on the part.

F.1.1.14 Define External Processes (A3114)*: Determines the required manufacturing processes that cannot be satisfied by resources available on the shop floor.

F.1.1.15 Define Gauges (A3115)*: Selects machine gauges required to perform inspection operations on the part.

F.1.1.16 Define Machine Instructions (A315)*: Produces the machine controller instructions required for the machines to perform operations on a given part.

F.1.1.17 Define Manufacturing Features (A2145): Captures design features of a part specified in the part information packet and represents them as manufacturing features in the digital part model.

F.1.1.18 Define Materials (A3116)*: Identifies the raw materials required to produce the part.

F.1.1.19 Define Modular and Dedicated Fixturing (A3112)*: Selects machine fixtures required to perform machining operations on the part.

F.1.1.20 Define Operator Data (A312)*: Produces graphical and textual instructions required by the operator to perform machining and inspection operations on a given part.

F.1.1.21 Define Part Properties (A2144): Captures non-graphical material and surface properties, and process information contained in the part information packet.

F.1.1.22 Define Resources (A311)*: Generates the list of items that must be available prior to the manufacturing process.

F.1.1.23 Define Route (A313)*: Produces the complete list of sequenced workstation types and cells including descriptions of the operations to be performed at each workstation.

F.1.1.24 Define Shop Floor Graphics (A3121)*: Creates graphical representation of the fixtures and tools necessary for visual instructions to perform machining operations on the part.

F.1.1.25 Define Shop Floor Operator Instructions (A3122)*: Describes the machine setup and machine tear-down instructions for a set of machining operations on a given part.

F.1.1.26 Define Workstations (A3113)*: Activity specifies candidate workstations machining operations for a given part are to be performed.

NOTE The workstations chosen are determined from the workstation capabilities (defined by the types of components and materials that can be processed, workstation limitations and ranges of operations) and availability.

F.1.1.27 Determine Production Cost & Schedule (A12)*: Delineates all costs incurred in the production of a part and the detailed agenda of task assignments and resource allocations to execute the manufacturing process.

F.1.1.28 Establish Part Acquisition Order (A11)*: Compiles customer part requisition information and initiates an order for work to begin manufacturing a particular part.

F.1.1.29 Execute Process Plan (A43)*: Carries out instructions specified in the process plan to manufacture a part.

F.1.1.30 Generate Manufacturing Data (A3)*: Produces information for use on the shop floor, defining the method to be used to manufacture a particular part, and feedback about the digital product data definition if necessary.

F.1.1.31 Generate Process Plan (A31)*: Creates process plans, and determines resources required for the process plan.

F.1.1.32 Generate Quality Reports (A44)*: Collects part quality data and inspection results generated during the execution of the process plan operations and assembles reports.

F.1.1.33 Generate Test and Inspection Plan (A3123)*: Generates the instructions for any final inspection or test procedures to be executed after a part has been manufactured and before the part is shipped.

F.1.1.34 Identify & Evaluate Design Problem (A33)*: Determines design exceptions discovered during creation of the process plan for a part.

NOTE Anomalies are evaluated for corrective action to be taken, if any; otherwise, the condition will warrant termination of the process plan generation.

F.1.1.35 Identify Base Shape (A212): Determines the basic part shape as the building block for part model creation.

EXAMPLE Examples of basic part shapes may be cylinder, cube, or prism.

F.1.1.36 Identify Manufacturing Features (A213): Recognizes and classifies manufacturing features to be applied to the base shape for part model creation.

F.1.1.37 Import ISO 10303 Data (A22): Reads in a ISO 10303 part definition and creates digital product data definition.

F.1.1.38 Maintain Shop Floor Resources (A132)*: Generates internal preventative maintenance requirements, and schedules and requests internal and external preventive and outage maintenance.

F.1.1.39 Manage Equipment & Materials (A13)*: Tracks and coordinates material and equipment usage, and maintenance.

F.1.1.40 Manage Inventory (A131)*: Tracks and acquires necessary materials and equipment needed for manufacturing.

F.1.1.41 Manage Manufacturing Process (A1)*: Provides administrative support for manufacturing mechanical parts from receipt of a customer part requisition through part shipment.

F.1.1.42 Manufacture Mechanical Parts (A0): Defines and processes all data and interfaces necessary for manufacturing machined parts.

F.1.1.43 Monitor Cutting Tool Inventory (A1311)*: Tracks and obtains necessary tools needed to support manufacturing.

F.1.1.44 Monitor Dedicated Fixture Inventory (A1314)*: Tracks and obtains necessary dedicated fixtures needed to support manufacturing.

F.1.1.45 Monitor Indirect Inventory (A1316)*: Tracks and obtains necessary indirect material items needed to support manufacturing. Indirect materials include both consumable and deliverable items.

F.1.1.46 Monitor Machine Inventory (A1315)*: Tracks and obtains necessary machines needed to support manufacturing.

F.1.1.47 Monitor Material Inventory (A1312)*: Tracks and obtains necessary raw stock and components needed to support manufacturing.

F.1.1.48 Monitor Modular Fixture Inventory (A1313)*: Tracks and obtains necessary modular fixtures needed to support manufacturing.

F.1.1.49 Operate Shop Floor (A4)*: Executes the work plans required for manufacturing a part.

F.1.1.50 Pack & Ship Finished Parts (A45)*: Coordinates packing and shipment of finished parts to the customer and receipt of shipment notifications.

F.1.1.51 Process Design Exception (A15)*: Evaluates a design problem encountered during the creation of a process plan and determines a recommendation for design problem resolution.

F.1.1.52 Process Discrepant Part (A14)*: Determines services needed to investigate the causes of inspection failure, process plan failure, or process plan execution failure, and defines the corrective action to be taken.

F.1.1.53 Produce Machine Code (A3151)*: Compute the numerical control instructions for machine controllers.

F.1.1.54 Refine Product Data Definition (A23): Corrects data definition errors and/or modifies/adds the information necessary to completely define the part for manufacturing.

F.1.1.55 Release & Track Shop Work Order (A16)*: Releases the schedule and tracking information required for manufacturing a part on the shop floor.

F.1.1.56 Track Shop Floor Utilization (A133)*: Monitors machine work loads.

F.1.1.57 Validate Part Model (A215): Validates the accuracy and completeness of the digital part model with respect to the part data definition from the part information packet.

F.1.1.58 Verify Product Data Definition (A24): Verifies that the product data definition is syntactically and semantically correct, and validated for data completeness.

F.1.1.59 Verify Machine Tool Path (A3152)*: Simulates the operations of a workstation where the results are reviewed and verified to determine whether or not the machine programs will function correctly.

F.1.2 Application activity model ICOM definitions

The following terms are used to define the inputs, controls, outputs, and mechanisms of the AAM diagrams.

F.1.2.1 Application Interpreted Model Schema: An EXPRESS schema that specifies the ISO 10303 constructs used for the communication of the information specified in the application reference model of an AP.

F.1.2.2 Basic Part Shape: The categorization of the shape of a given part as determined from the part technical data that is used as the basis for machining feature application.

F.1.2.3 Bill of Materials:* A definition of all materials, parts, components and special tools necessary to manufacture a given part.

F.1.2.4 Computer Integrated Manufacturing: Hardware and software technology to aide in the manufacturing of a part.

EXAMPLE Computer-aided process planning (CAPP), Computer-aided manufacturing (CAM) and Computer-aided design (CAD) are examples of computer integrated manufacturing.

F.1.2.5 Corrective Action Plan:* Process adjustment instructions that are required to correct a shop floor operation problem.

F.1.2.6 Customer Order: Notification that data has been received from the customer necessary to process a request for the manufacture of a part.

F.1.2.7 Cutting Tools:* A resource required for material removal in machining operations as defined in the process plan.

F.1.2.8 Cutting Tool Requisition: An order for tools required to support part manufacturing.

F.1.2.9 Dedicated Fixture Requisition: An order for dedicated fixtures required to support part manufacturing.

F.1.2.10 Design Exception Notice: The notification of a design discrepancy discovered during the creation of the process plan for a given part.

EXAMPLE A tolerance called out on a feature may not be held by a machine.

F.1.2.11 Design Specifications: Part design requirements that manufacturing process must follow or adhere.

F.1.2.12 Digital Product Data Definition: The set of data elements that completely defines a given part represented in a computer interpretable form.

F.1.2.13 Digital Product Data Schema: A view of a physical representation of product data on a computer. The schema encodes the semantics that completely define the product for manufacturing in some computer interpretable syntax.

F.1.2.14 Digital Product Data Work Order: Notification that the data needed to create the digital product data definition for a part has been collected.

F.1.2.15 Electrical Property Specifications: Document containing part electrical characteristic requirements that manufacturing process must follow or adhere.

F.1.2.16 Engineering Service Request:* A description of a problem encountered during the execution of the process plan and a request for determination of the problem cause and the development of corrective actions or corrective services.

F.1.2.17 Equipment and Materials Requisition: An order for items required to support part manufacturing.

F.1.2.18 Equipment Requirements: Needs for items to produce a part as determined by process planning.

F.1.2.19 Estimated Process Time:* Time measurements for specific operations at a workstation determined from approved standard operation times for that workstation.

F.1.2.20 External Processes:* Specified processes defined for a given part that are performed outside the shop that the part is being machined.

F.1.2.21 Featured Part Shape: The representation of the shape of a part defined by manufacturing form features.

F.1.2.22 Gauges:* A resource required for part measurements as defined in the process plan.

F.1.2.23 Geometric/Topological Shape Representation: Part shape as defined by geometric and topological data elements.

F.1.2.24 Indirect Stock Requisition: An order for indirect items required to support part manufacturing. Indirect materials include both consumable and deliverable items.

F.1.2.25 Inspection Results:* Part inspection information generated from the actual manufacturing execution of process plan operations.

F.1.2.26 Inspection Specifications: Document containing part inspection requirements that manufacturing process must follow or adhere.

F.1.2.27 Inventory Status:* An availability check of inventory on order and on hand.

F.1.2.28 ISO 10303 Physical File Definition: A definition of a file that conforms to ISO 10303-21 containing ISO 10303 product data that is to be exchanged between computer systems.

F.1.2.29 Machine Code:* Machine instructions required by a machine tool controller to automatically control the machining of a given part.

F.1.2.30 Machine Code Feedback:* Instructions that define how to modify numerical control (NC) code that did not pass verification analysis and requires further refinement.

F.1.2.31 Machine Instructions:* Commands that are sent to a machine to execute a task. Typically machine instructions are in the form of NC programs to be downloaded to the machine controller or a machine tool. Directions in a form understandable by a machine that produce the desired transformations of an input material into an intermediate or final form.

F.1.2.32 Machine Requisition: An order for machines required to support part manufacturing.

F.1.2.33 Machining Features: A shape that conforms to some preconceived pattern for an application purpose.

F.1.2.34 Maintenance Schedule:* An agenda of maintenance and repair activities for the shop floor.

F.1.2.35 Manufacturing Practices:* Procedures pertaining to the manufacturing process that provide intuitive knowledge about certain aspects of manufacturing a part that are generally accepted as standard within the manufacturing industry.

F.1.2.36 Materials:* A definition of the raw stock as called out in the process plan.

F.1.2.37 Material Properties: Characteristics of the part materials defined in the part technical data.

F.1.2.38 Material Requirements: Information about the material required to produce the part as interpreted from the technical data packet.

F.1.2.39 Material Requisition: An order for materials required to support part manufacturing.

F.1.2.40 Material Specifications: Document containing part material requirements that manufacturing process must follow or adhere.

F.1.2.41 Mechanical Part:* A thing of substance produced from raw materials. A mechanical part contains physical material characteristics.

F.1.2.42 Modular and Dedicated Fixtures:* A resource required to secure a given part for machining operations as defined in the process plan.

F.1.2.43 Modular Fixture Requisition: An order for modular fixtures required to support part manufacturing.

F.1.2.44 Operator Data:* Both the graphical and textual data supplied to an operator on the shop floor.

F.1.2.45 Optical Property Specifications: Document containing part optical characteristic requirements that the manufacturing process must follow or adhere.

F.1.2.46 Packaged Part:* A mechanical part that has been packaged and prepared for shipping to a customer.

F.1.2.47 Part Acquisition Order: Notification that Manufacturing data needed for shop floor processing has been collected.

NOTE This data includes customer order information, part identification, quantity, and schedule dates.

F.1.2.48 Part Administration Data: Information about the part that is used to make decisions concerning the parts status within the manufacturing process from a management prospective.

F.1.2.49 Part Data Definition: Information that completely describes a part to be manufactured.

F.1.2.50 Part Dimensions: The measured values associated with a physical part that defines the shape and location characteristics of part features. There may be two types of measurement - size dimensions and location dimensions. Size dimensions give the size of a piece, component part, hole, slot or other feature. Location dimensions fix the relationship of a component part (projections, holes, slots, and other significant forms) or a piece or structure.

F.1.2.51 Part Exception Notice:* The notification of a discrepancy in part processing on the shop floor. This can be a failure of the process plan or its execution.

F.1.2.52 Part Information Packet:* Part technical data and customer order information.

F.1.2.53 Part Model: The complete representation of a given part that captures part design aspects (including all elements that define a part and their relationships) as interpreted from the technical data packet.

F.1.2.54 Part Model Feedback: Information about the part model that is a result of its verification.

F.1.2.55 Part Pedigree:* Complete history of processing of a part information packet through the manufacturing process to the creation of the final part.

F.1.2.56 Part Pedigree Data:* Any particular piece of data that comprises the complete history of the part.

F.1.2.57 Part Shape Form Feature Representation: The set of design form feature types as determined from the part technical data to be applied to the selected basic part shape in the creation of the part model.

F.1.2.58 Part Technical Data: The physical, material, and manufacturing process information that describes a given part.

F.1.2.59 Part Tolerances: Part dimension variances used in the manufacture of a given part.

NOTE Tolerances impact the types of manufacturing processes capable of producing particular part features. Part tolerances are added to the part form features.

F.1.2.60 Part Workstation History:* Data that represents workstation part transactions resulting from process plan execution.

NOTE Part workstation history reports machine, tool, and fixture usage, material identification, and material certification.

F.1.2.61 Pedigree Creation Request: A notice that signifies that a given part has been identified for part pedigree tracking.

F.1.2.62 Process Plan:* The data that provides the available machines with the engineering specifications required to manufacture a given part.

NOTE This data includes resources, operator data, routing, process times, bill of materials, and machine instructions.

F.1.2.63 Process Properties: Characteristics of manufacturing processes for a given part as defined in the part technical data.

F.1.2.64 Process Specifications: Document containing part process requirements that manufacturing process must follow or adhere.

F.1.2.65 Product Data Definition Completion Notice:* A notification that the digital product data definition for a given part is complete.

F.1.2.66 Product Data Definition Deficiency Notice:* A notification that the digital product data definition for a given part is incomplete.

F.1.2.67 Product Data Definition Feedback: Information about the data elements defining a part that did not pass verification analysis and requires further refinement.

F.1.2.68 Product Data Definition Status Notice:* A notification that the digital product data definition for a given part is either complete or incomplete.

F.1.2.69 Production Cost & Schedule:* All the information that describes the cost to produce a given part, and the calendar for all events associated with the manufacture of that part.

F.1.2.70 Property Requirements: Part characteristics specified in the part technical data.

F.1.2.71 Quality Reports:* A formatted compilation of inspection results, resource usage and condition results, and quality data.

F.1.2.72 Raw Stock:* The unprocessed materials necessary to manufacture a given part or set of parts.

F.1.2.73 Released Shop Work Order: The shop work order that is formally released to the shop floor when all items and process plans are available, and shop capacity permits.

F.1.2.74 Request for Technical Data:* An inquiry made to obtain part technical data to manufacture a given part.

F.1.2.75 Resources:* All items required for the manufacture of a given part.

EXAMPLE Such items may include tools, fixtures, workstations, gauges, materials, and processes.

F.1.2.76 Resource Availability:* A report containing the status of obtainable resources at particular point in time.

F.1.2.77 Resource Usage and Condition:* The status of those resources consumed and the condition of those resources utilized during the execution of the process plan.

F.1.2.78 Routing:* A sequential list of manufacturing processes that are required to manufacture a given part.

F.1.2.79 Shop Floor Graphics:* Pictorial aids that are useful in explaining the machining processes to shop floor personnel by providing a snapshot view of a given part.

F.1.2.80 Shop Floor Operator Instructions:* Detailed textual descriptions that explain the machining processes to shop floor personnel.

F.1.2.81 Shop Work Order: Notification to the shop floor that part acquisition order data and process plan data defining the manufacturing requirements for a part is ready for processing.

F.1.2.82 Specifications: Documents containing requirements that the manufacturing process must follow or adhere.

F.1.2.83 Standards: Documents that establish engineering and technical limitations and applications for items, materials, processes, methods, designs, and engineering practices that have achieved formal consensus within a particular organizational context.

F.1.2.84 Surface Finish Properties: Part surface characteristics that define some required value of surface roughness as contained in the part technical data.

F.1.2.85 Surface Finish Specifications: Document containing requirements for the state of part surfaces that the manufacturing process must follow or adhere.

F.1.2.86 Technical Recommendation: The recommended resolution to a design problem discovered during the creation of the process plan for a part.

F.1.2.87 Test and Inspection Plan:* Instructions for any final inspection or test procedures to be executed after a part has been manufactured and before the part is shipped.

F.1.2.88 Time Standards:* Historical data for specific manufacturing resource configurations containing times spent to complete various manufacturing processes that is used for future process time estimations.

F.1.2.89 Verified Digital Product Data Definition: The set of complete data elements defining a part that is syntactically and semantically correct.

F.1.2.90 Verified Machine Code:* NC code that is validated for a particular machine and tool.

F.1.2.91 Validated Part Model: A part model that has been checked against the original part technical data to insure correctness and completeness.

F.1.2.92 Updated Part Acquisition Order: A revised part acquisition order that reflects additional information necessary for discrepant part processing.

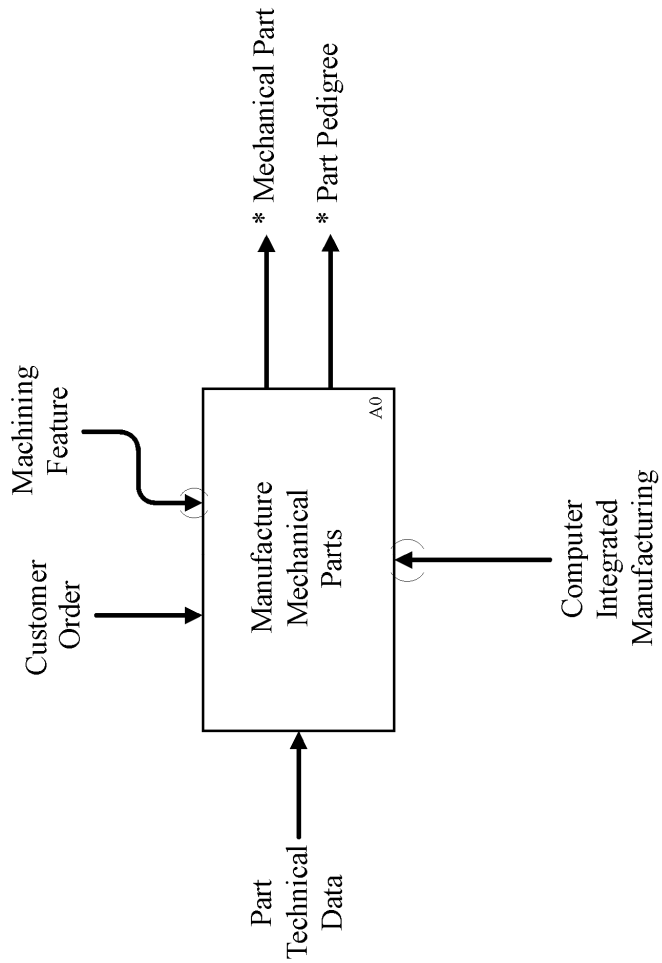
F.1.2.93 Workstations:* The machine center assigned to perform a particular operation on a part.

F.1.2.94 Workstation Control Commands:* All information describing operation, resource, process, and transportation requirements and instructions for a production operation to be performed on a part.

F.1.2.95 Work Load Report:* Information that identifies the utilization of shop floor resources.

F.2 Application activity model diagrams

The application activity model is given in figures F.2 through F.14. Activities and data flows tat are out of scope are marked with asterisks.



* - Indicates ICOMs and activities out of scope

Figure F.2 - Mechanical products definition for process planning using machining feature

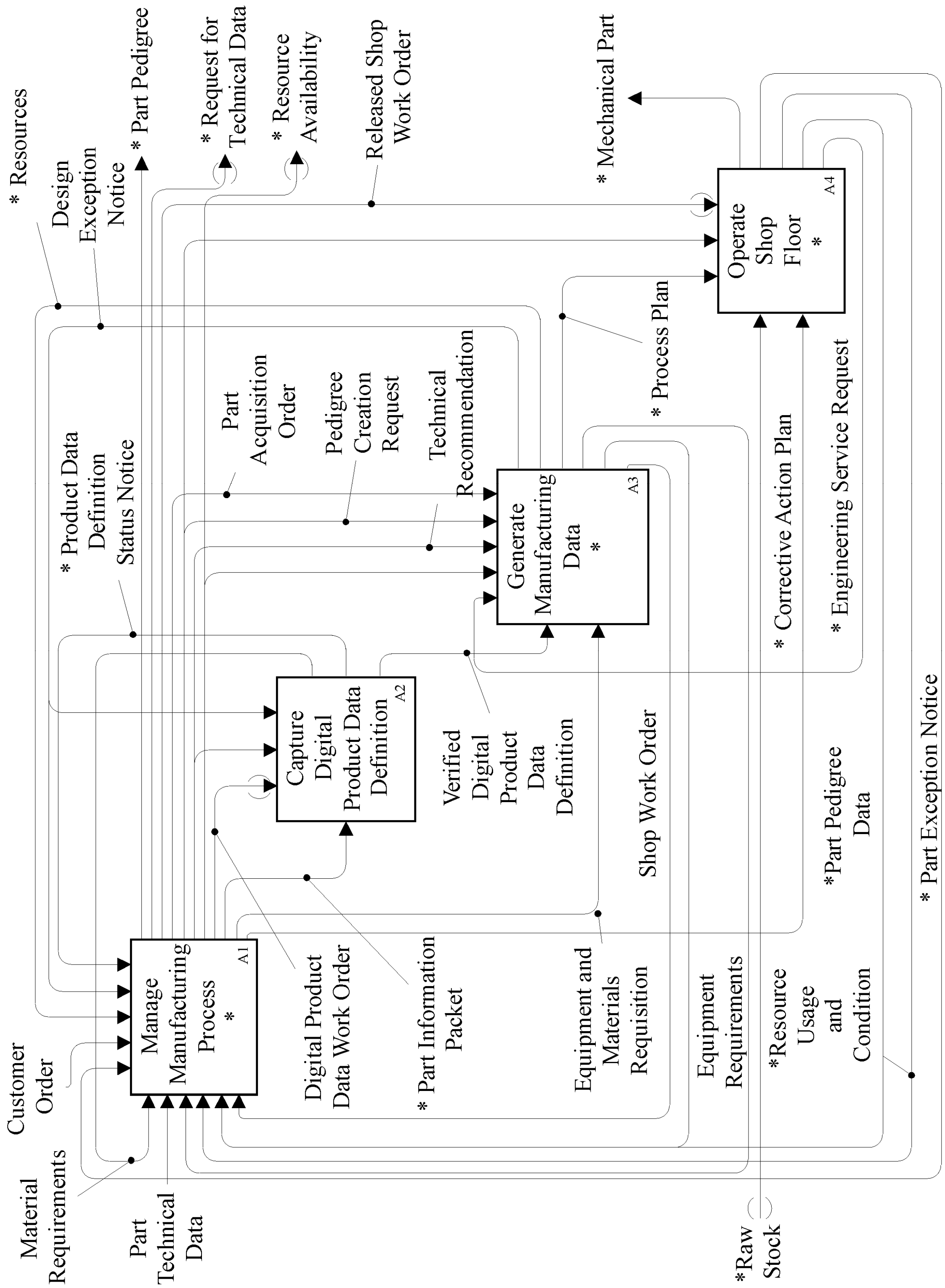


Figure F.3 - A0 manufacture mechanical parts

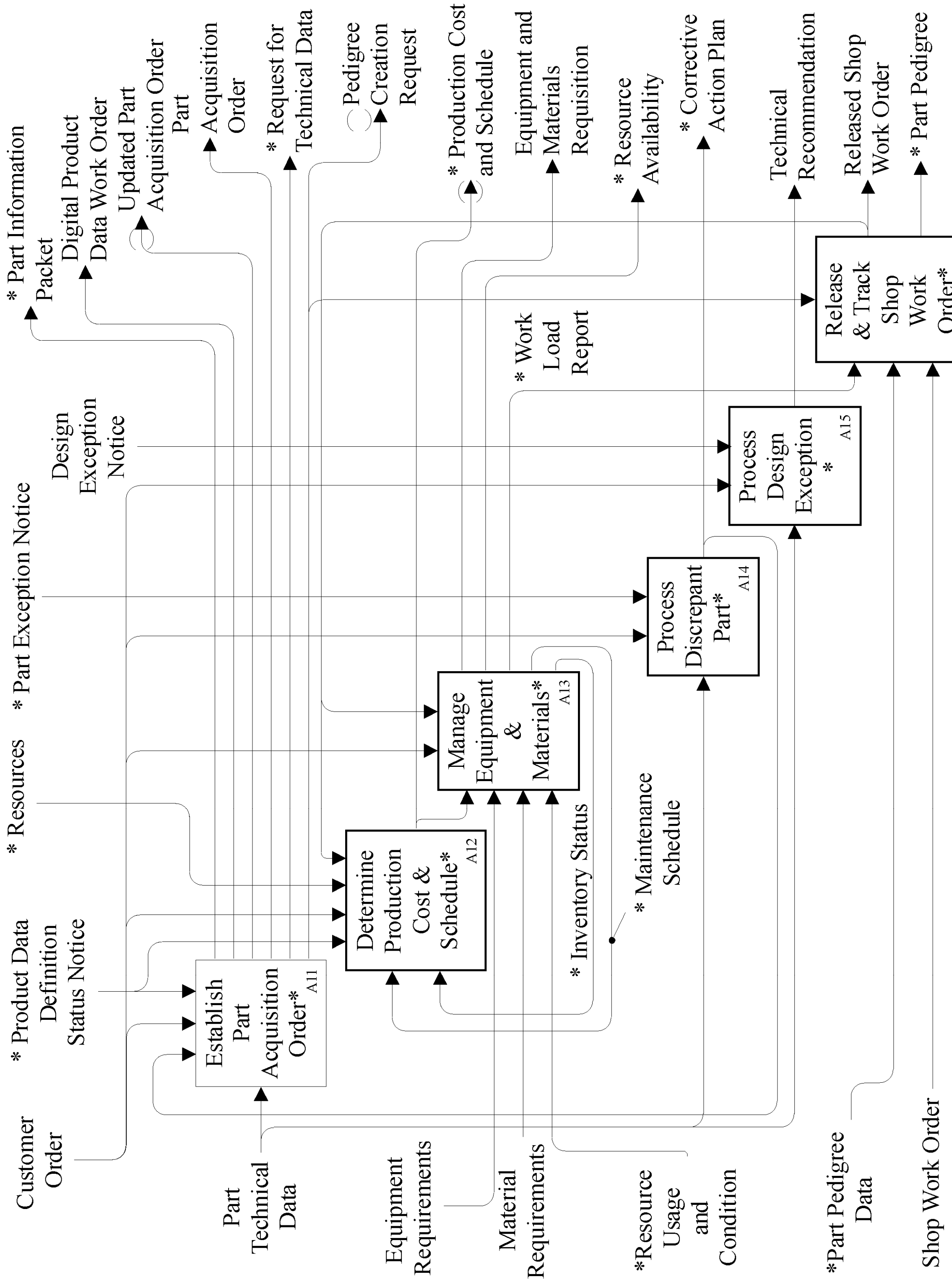


Figure F.4 - A1 manage manufacturing process

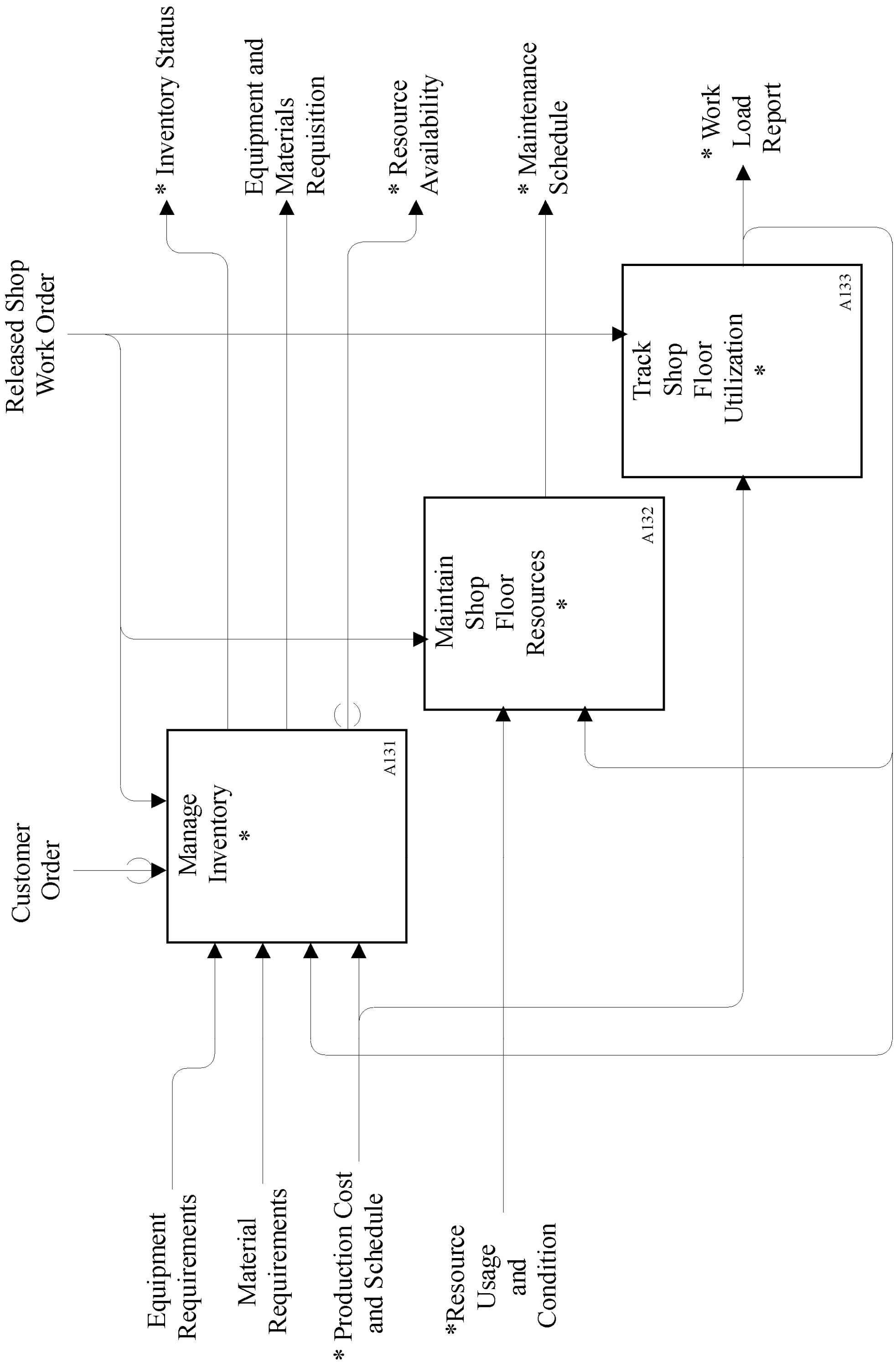


Figure F.5 - A13 manage equipment and materials

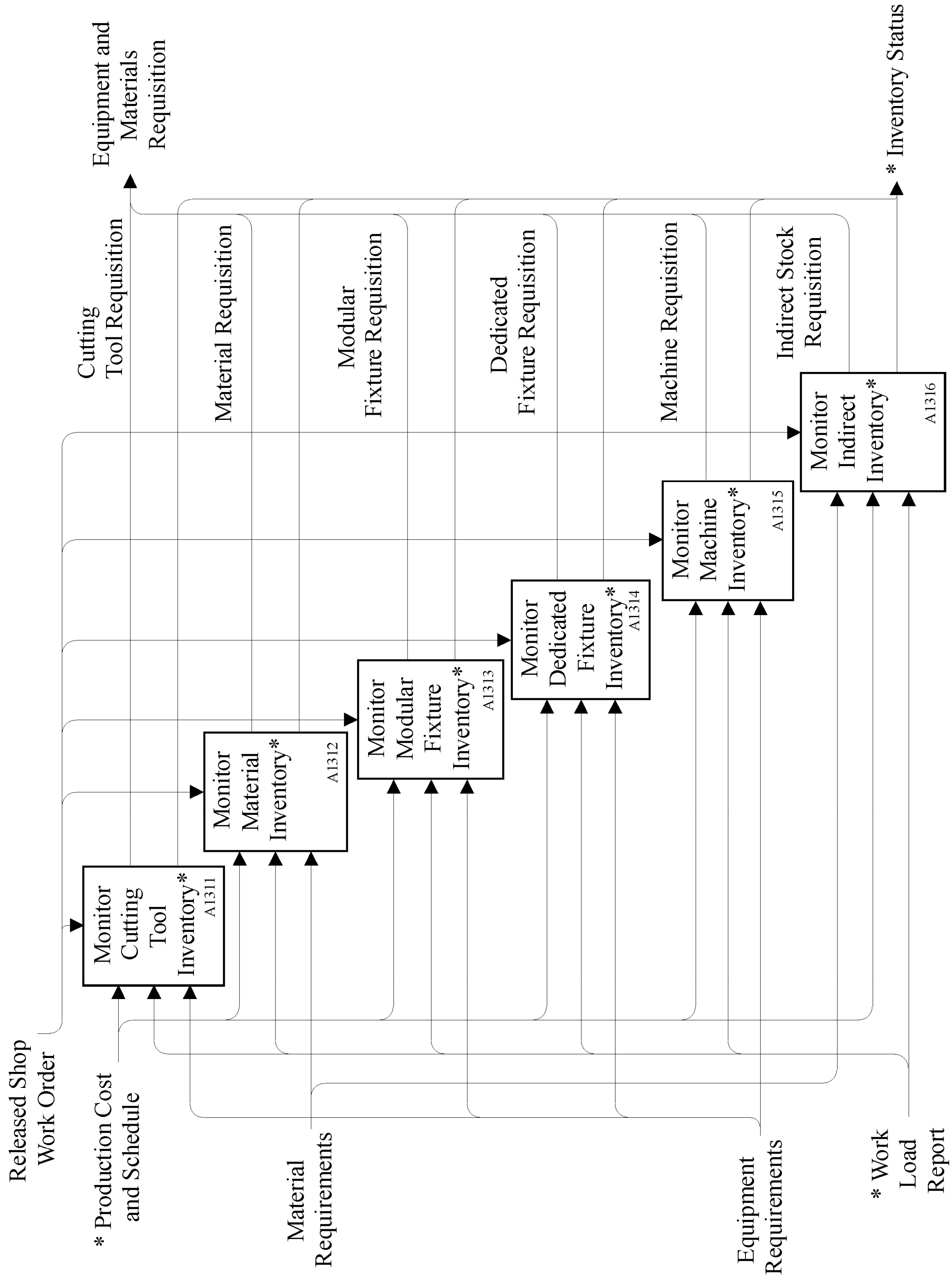


Figure F.6 - A131 manage inventory

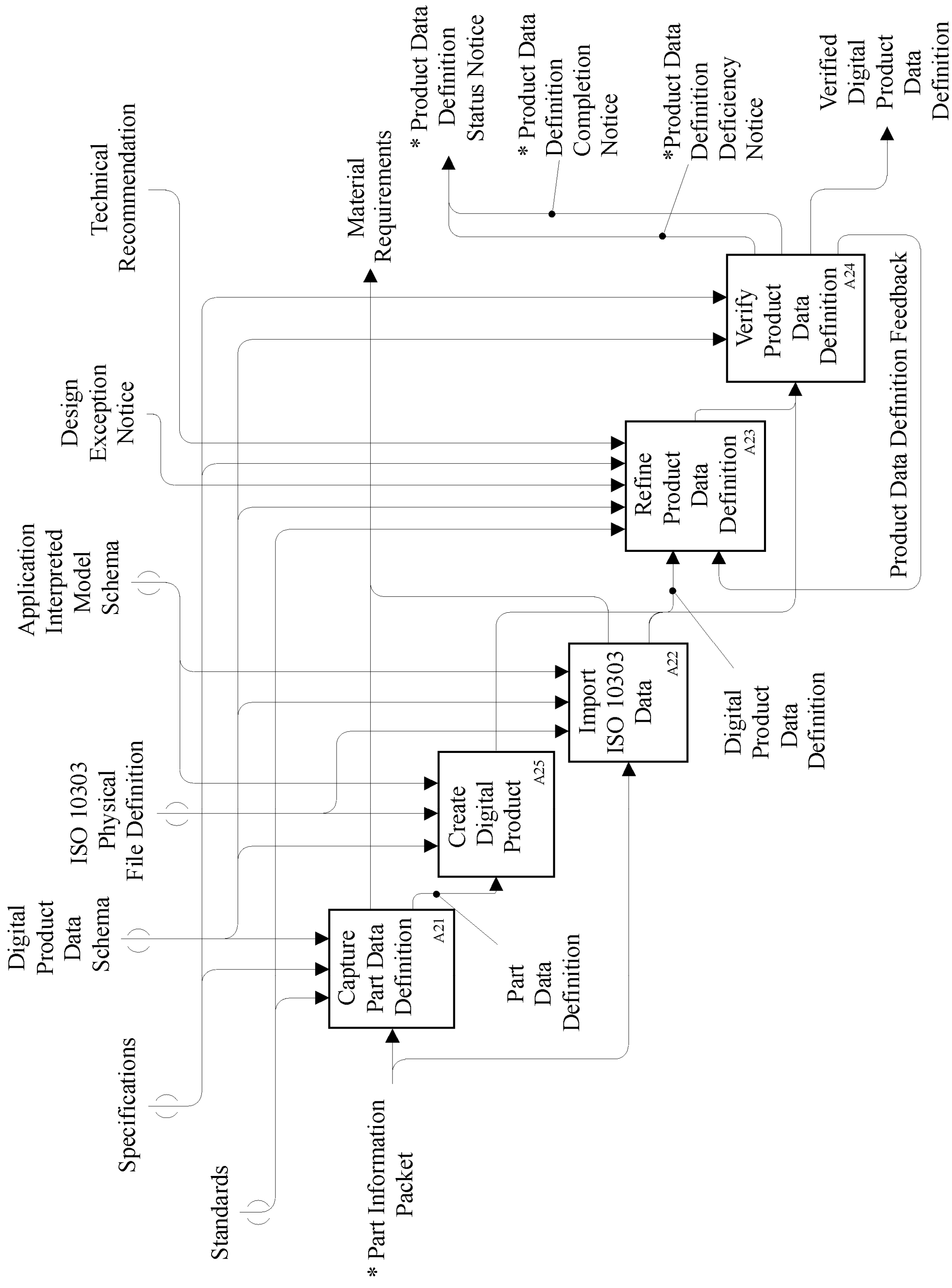


Figure F.7 - A2 capture digital product definition

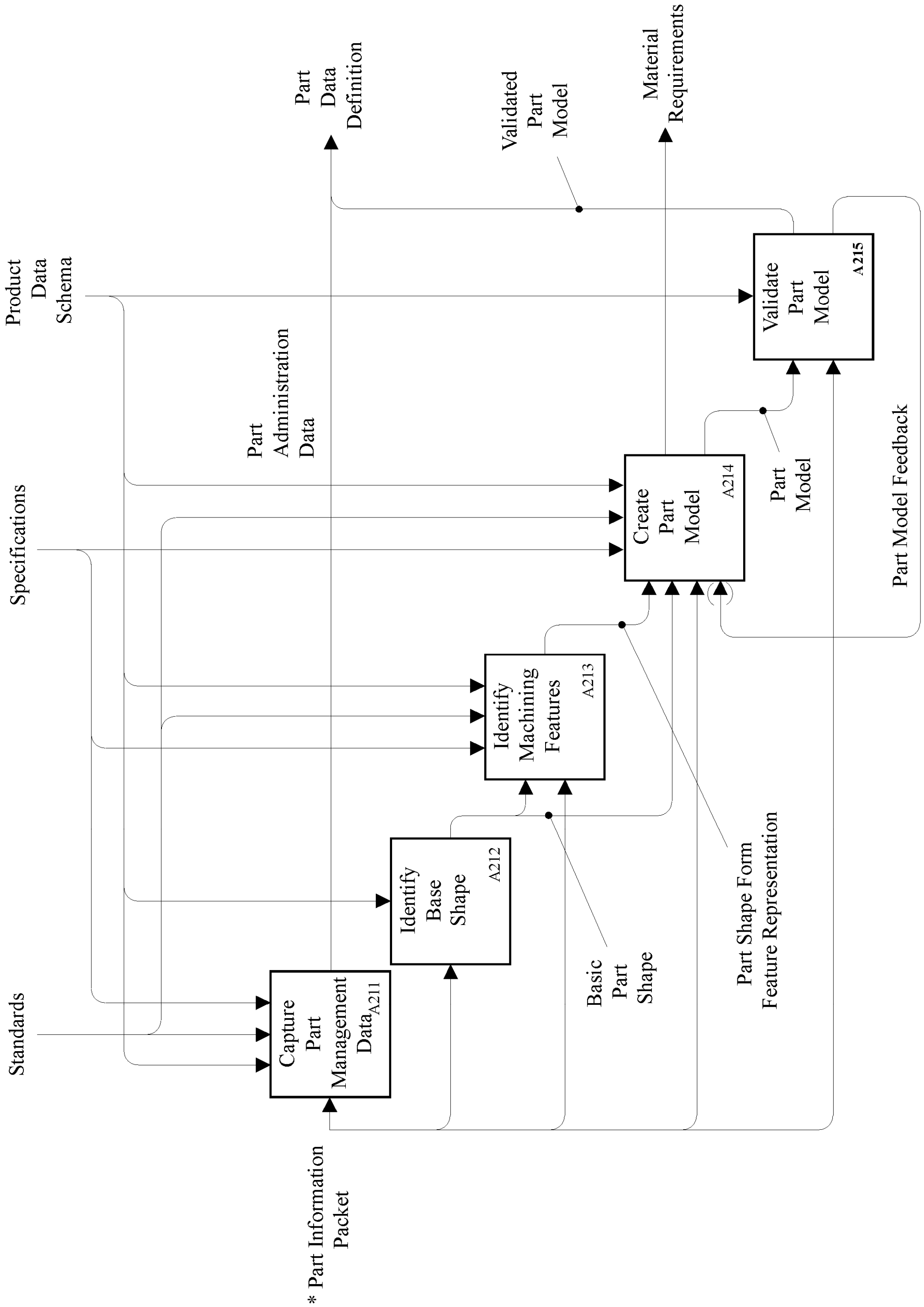


Figure F.8 - A21 capture part data definition

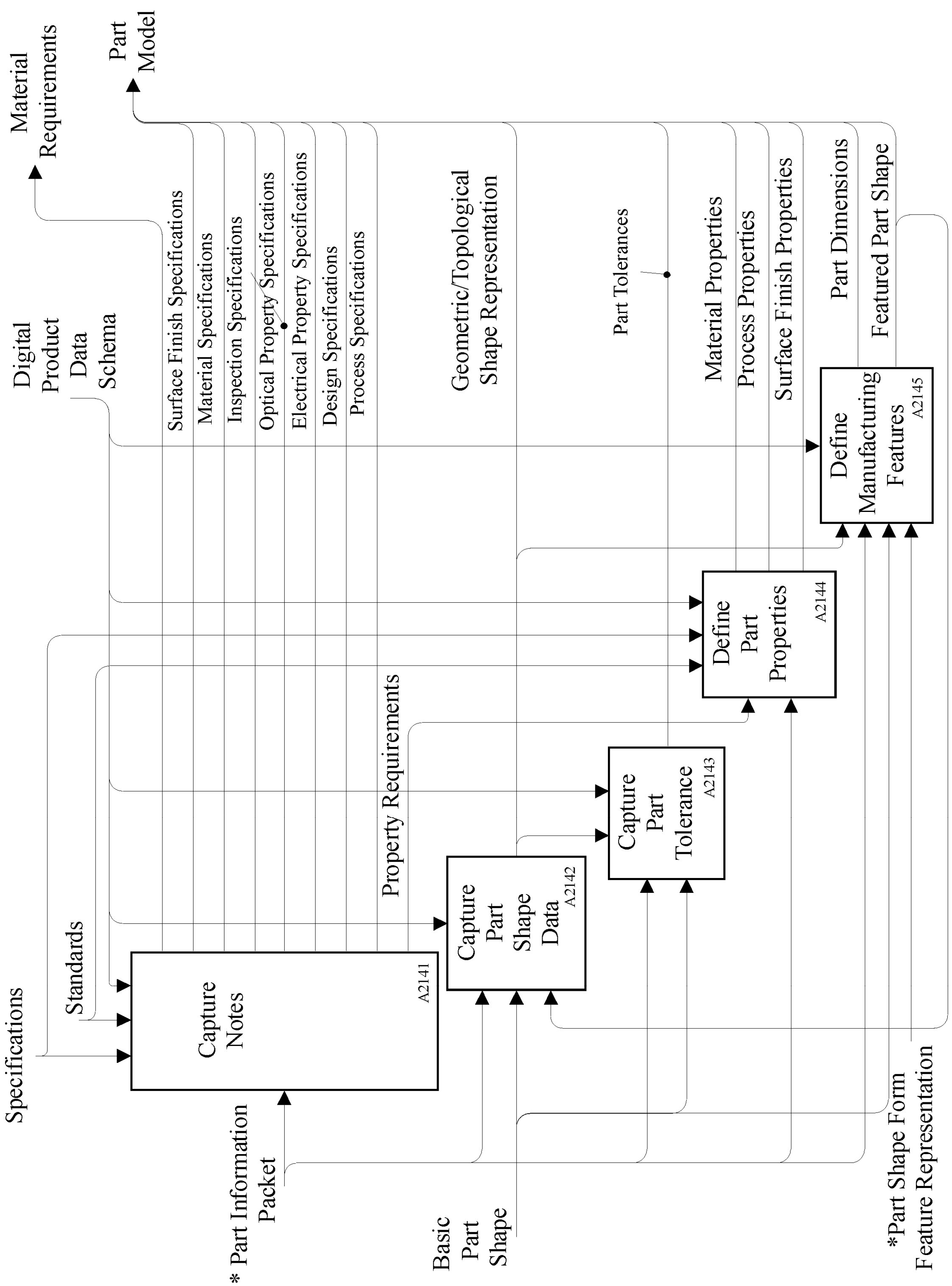


Figure F.9 - A214 create part model

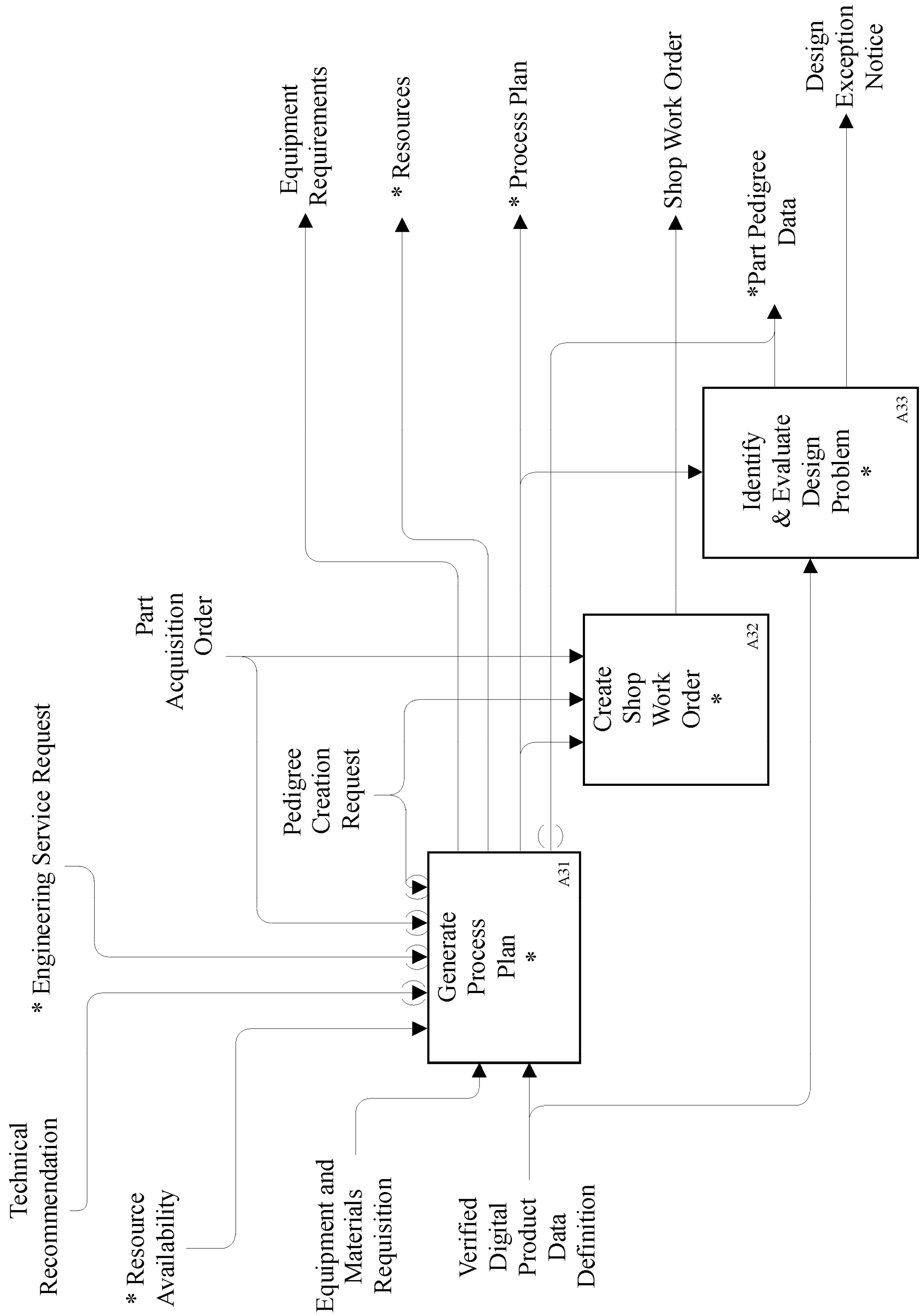


Figure F.10 - A3 generate manufacturing data

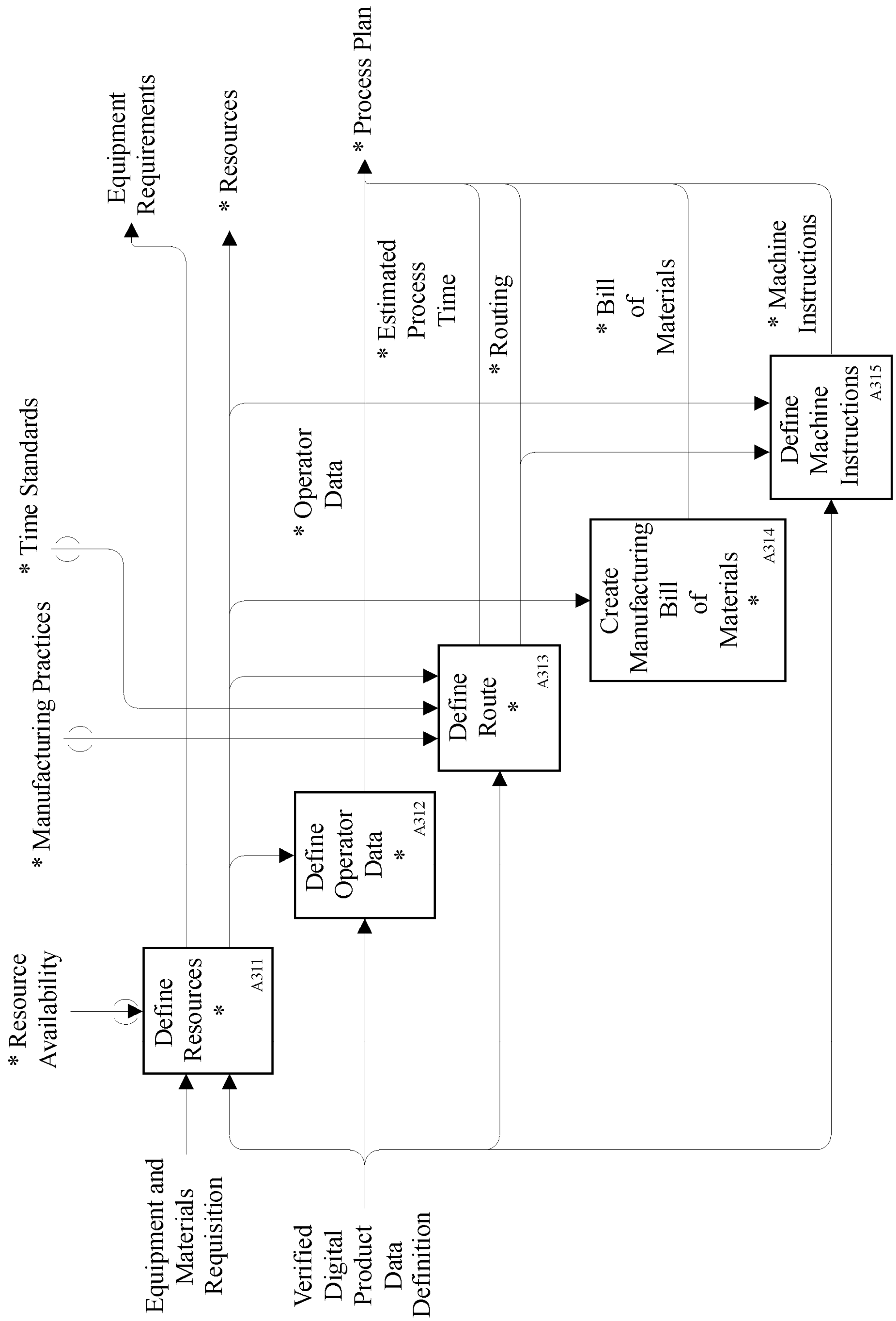


Figure F.11 - A31 generate process plan

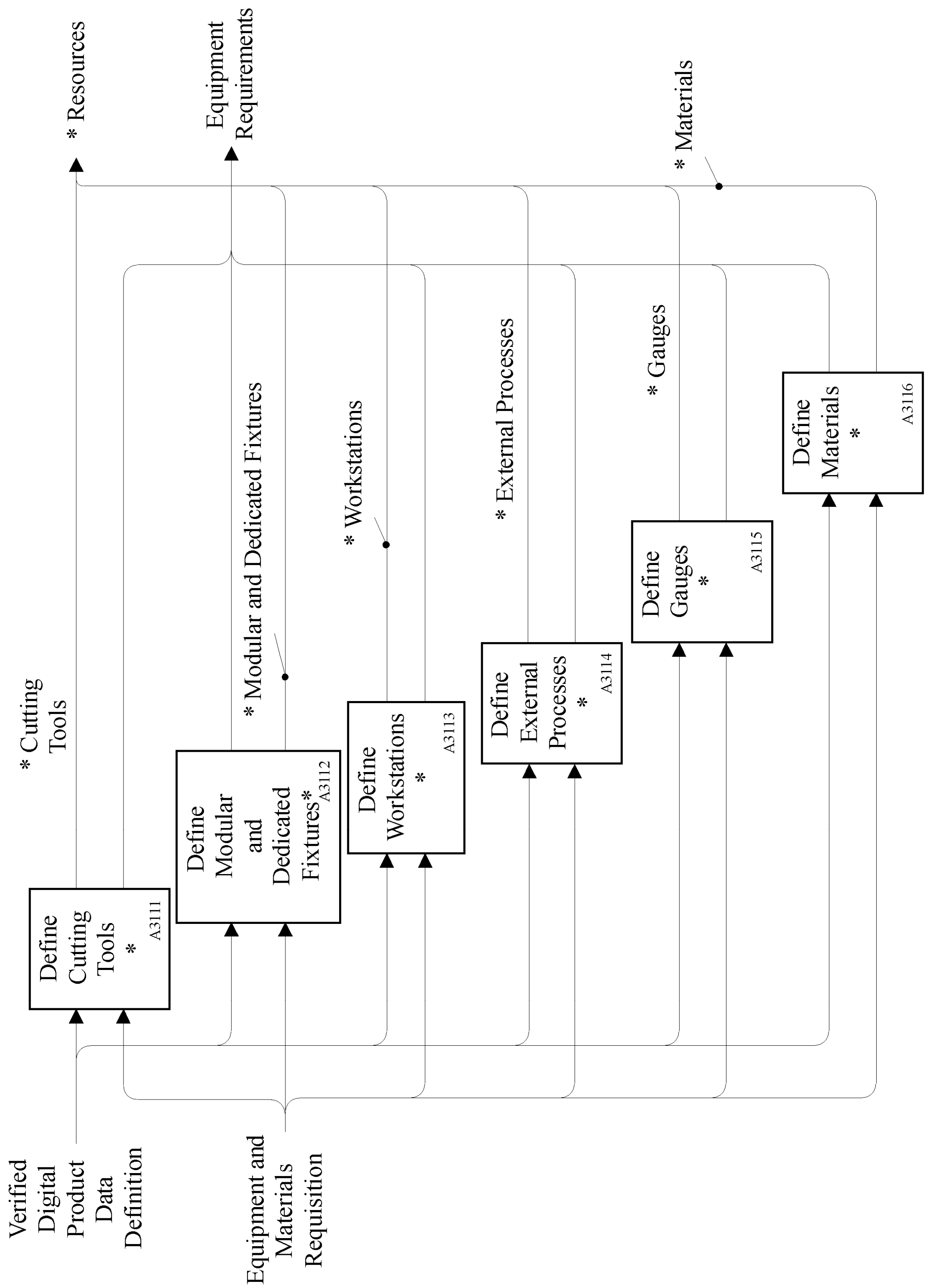


Figure F.12 - A311 define resources

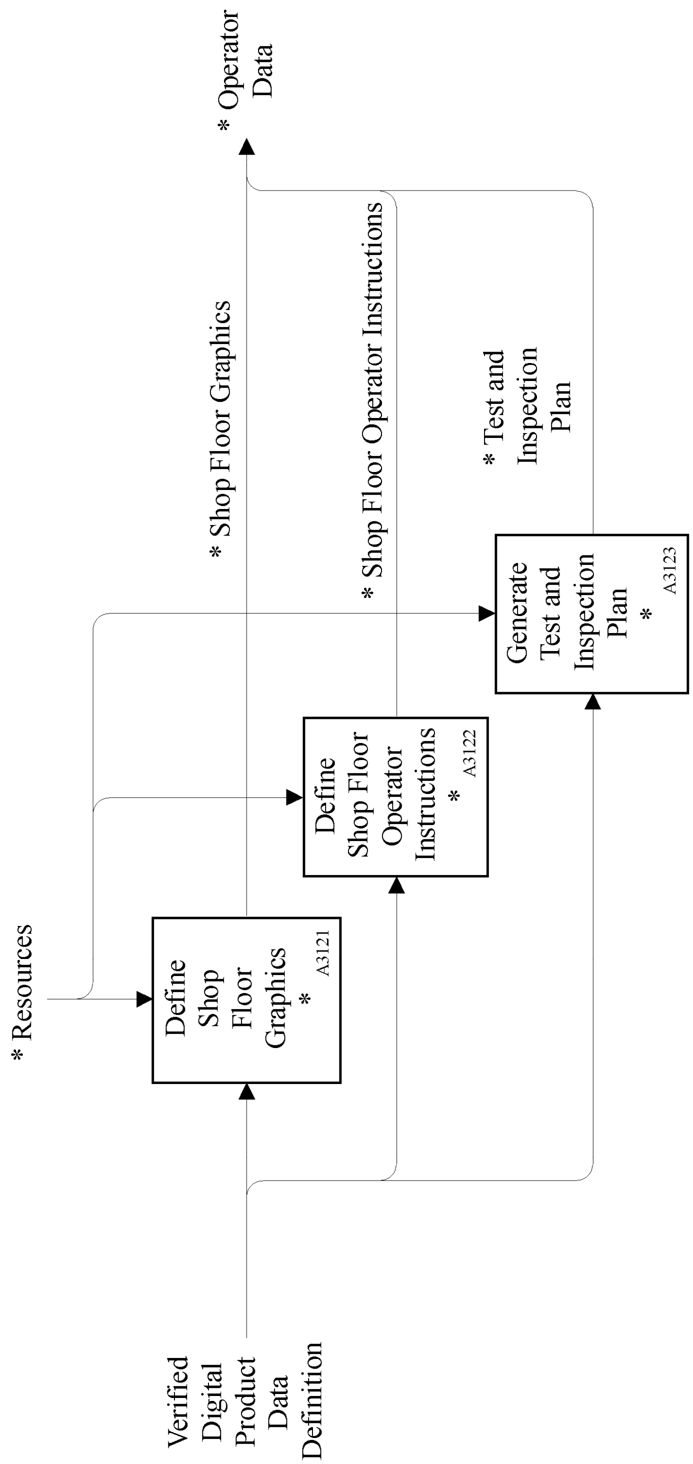


Figure F.13 - A312 define operator data

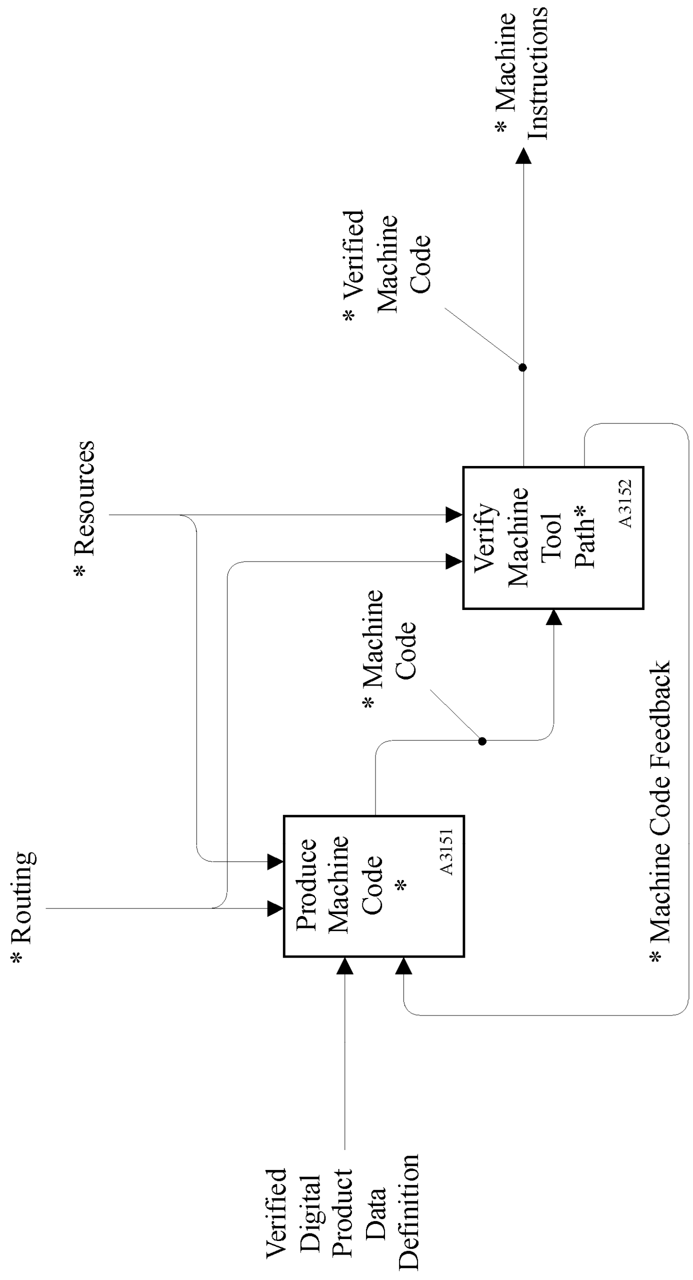


Figure F.14 - A315 define machine instructions

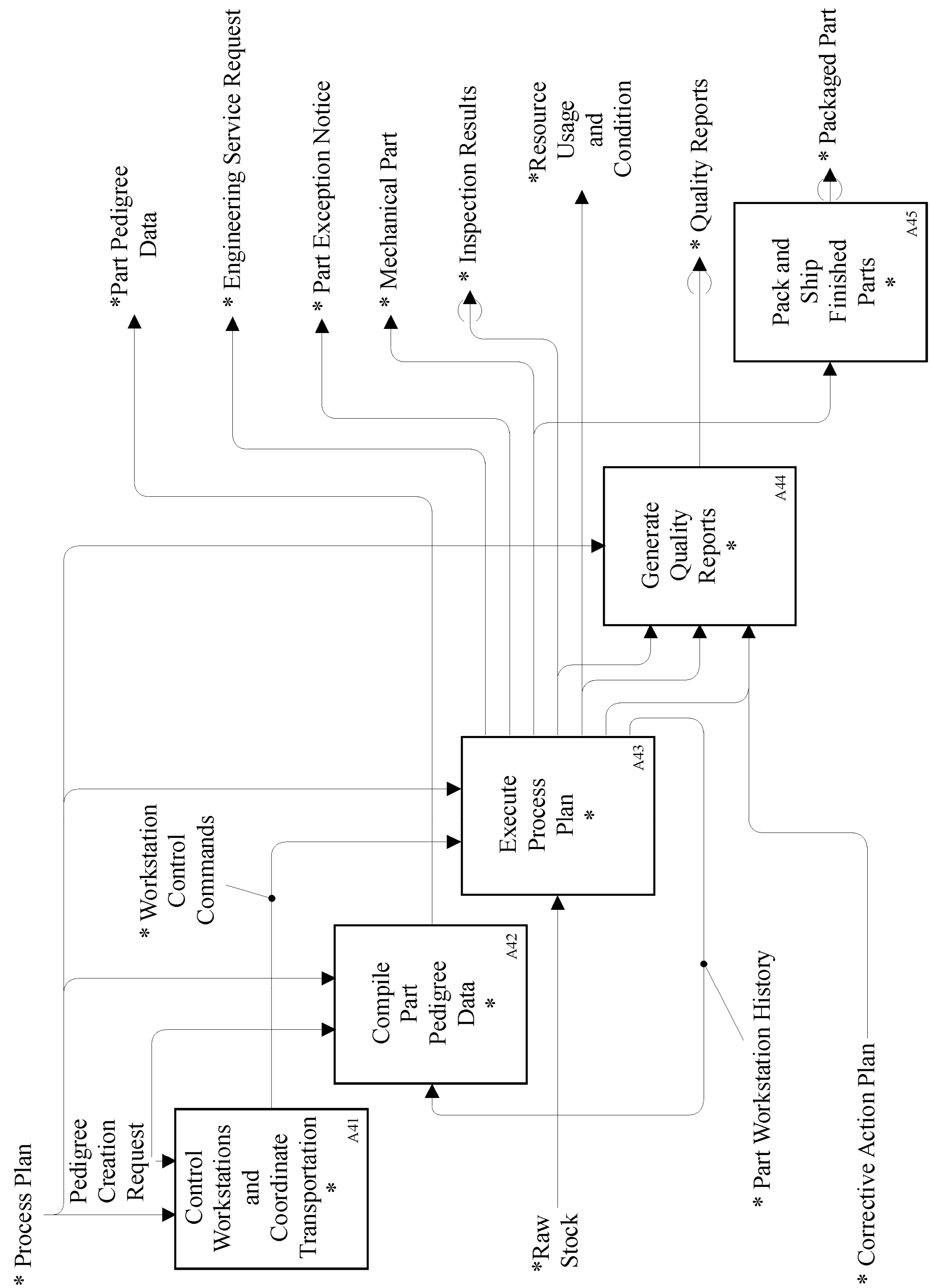


Figure F.15 - A4 operate shop floor

Annex G
(informative)
Application reference model

This annex provides the application reference model for this part of ISO 10303 and is given in figures G.1 through G.30. The application reference model id a graphical representation of the structure and constraints of the application objects specified in Clause 4. The graphical form of the application reference model is presented in the EXPRESS-G. The application reference model is independent from any implementation method.

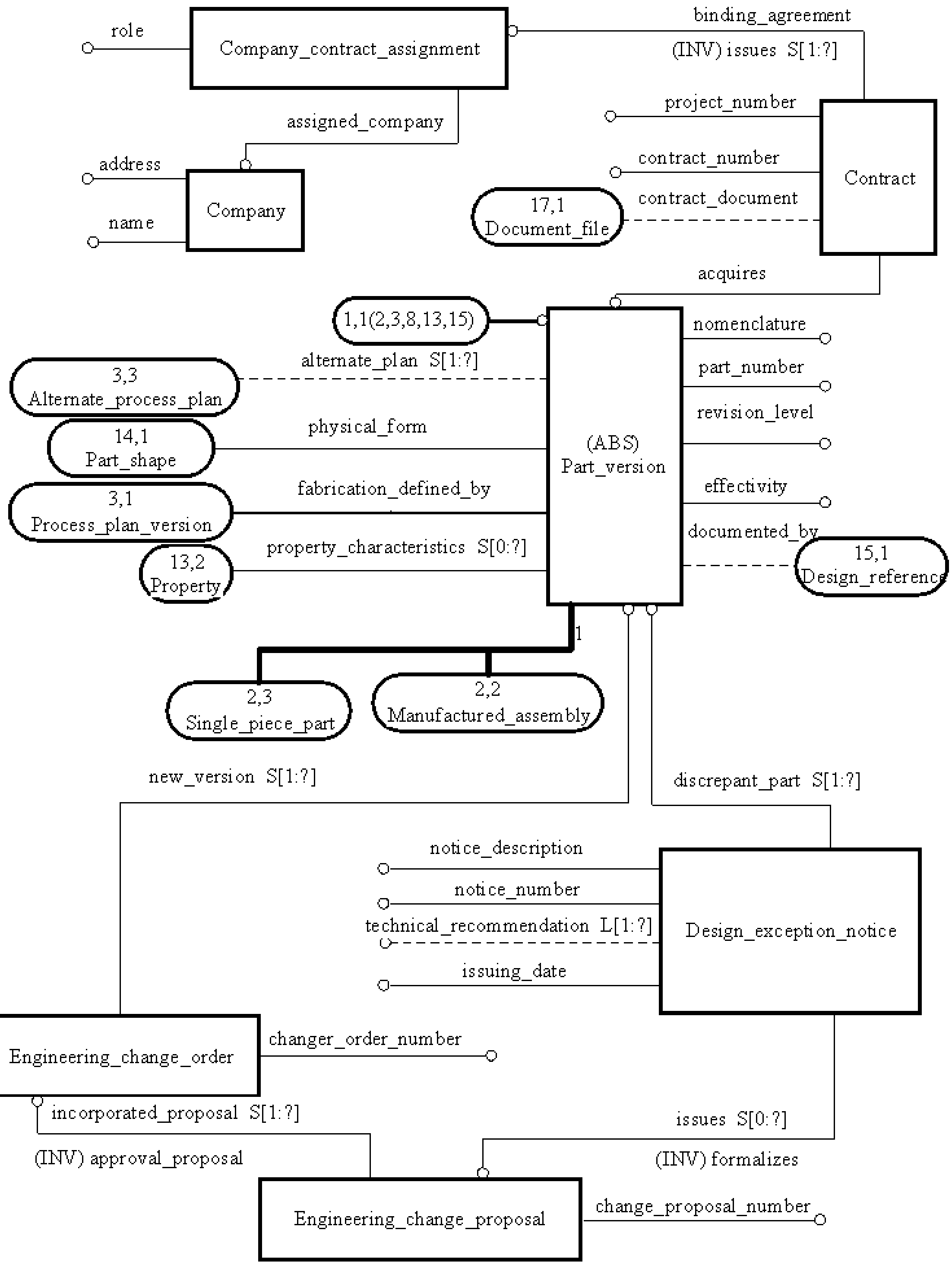


Figure G.1 — ARM EXPRESS-G diagram 1 of 30

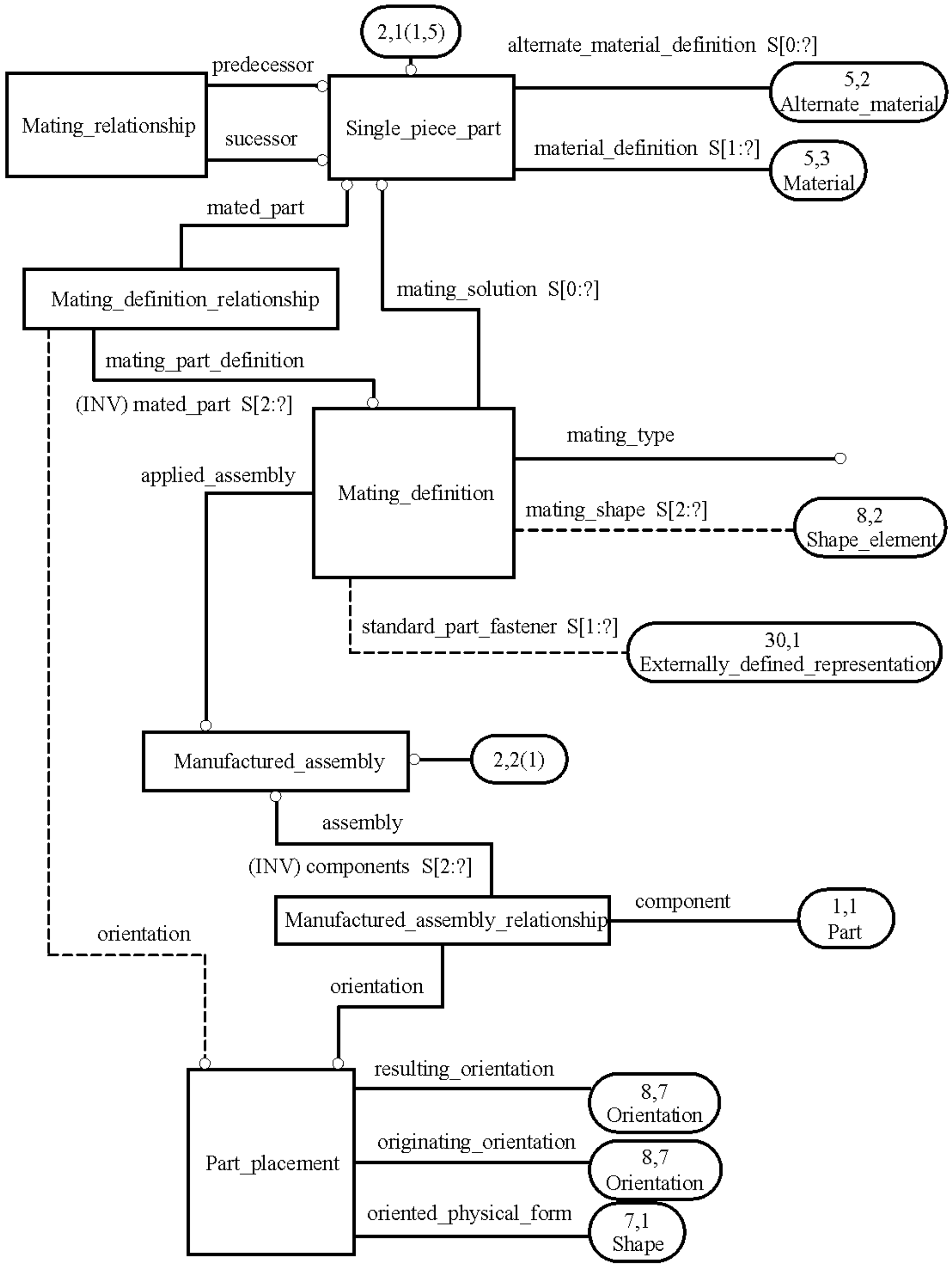


Figure G.2 — ARM EXPRESS-G diagram 2 of 30

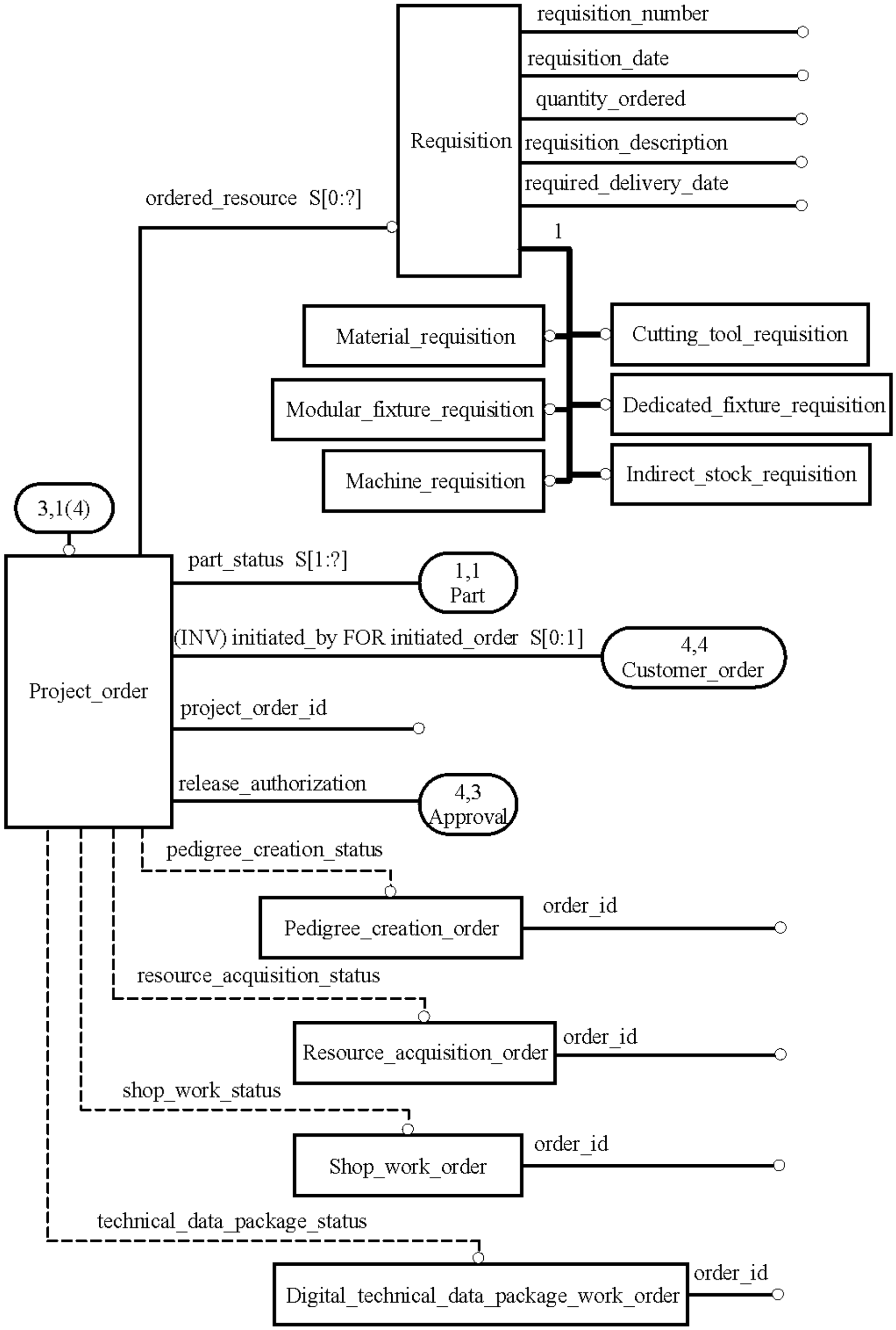


Figure G.3 — ARM EXPRESS-G diagram 3 of 30

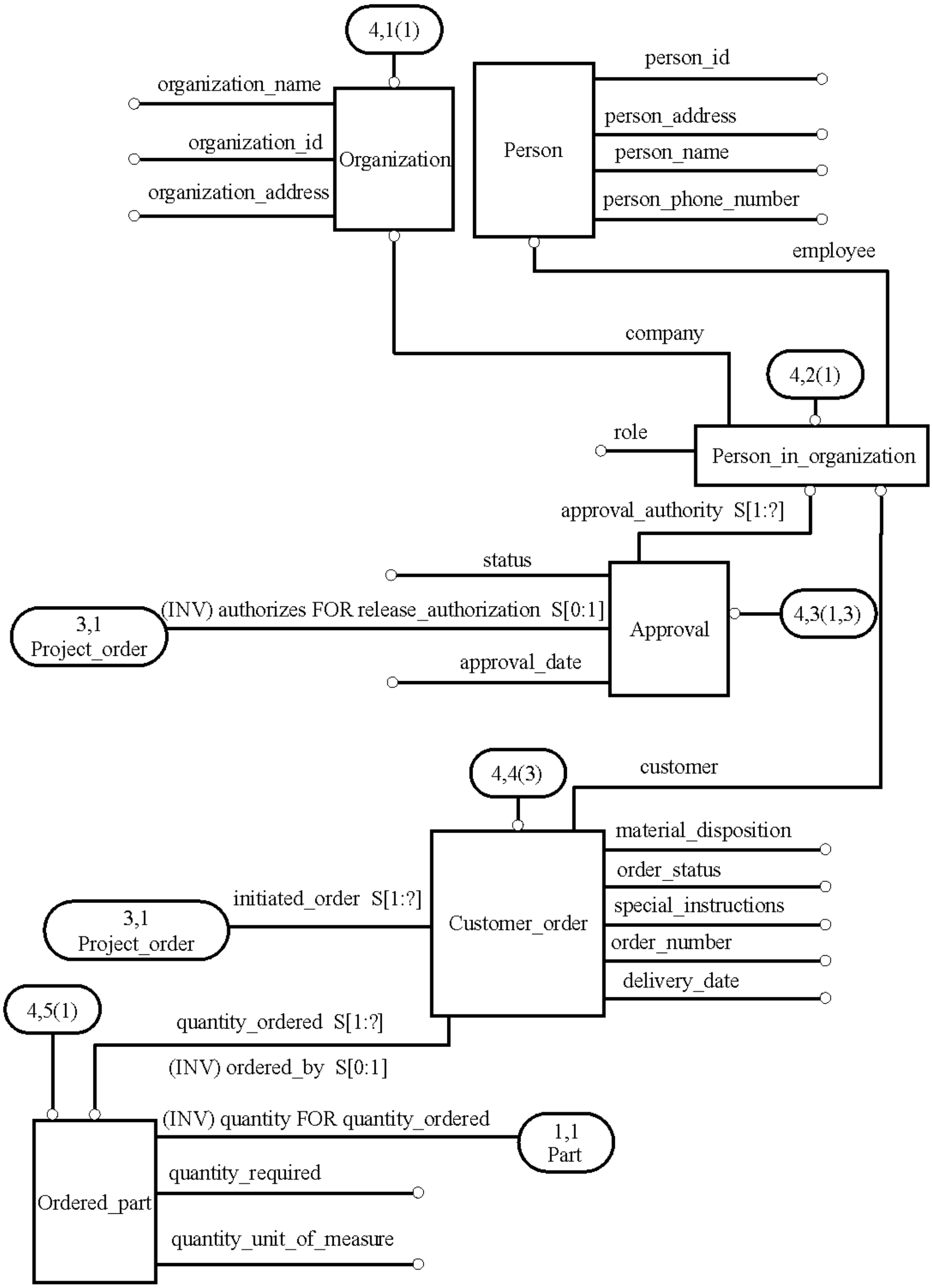


Figure G.4 — ARM EXPRESS-G diagram 4 of 30

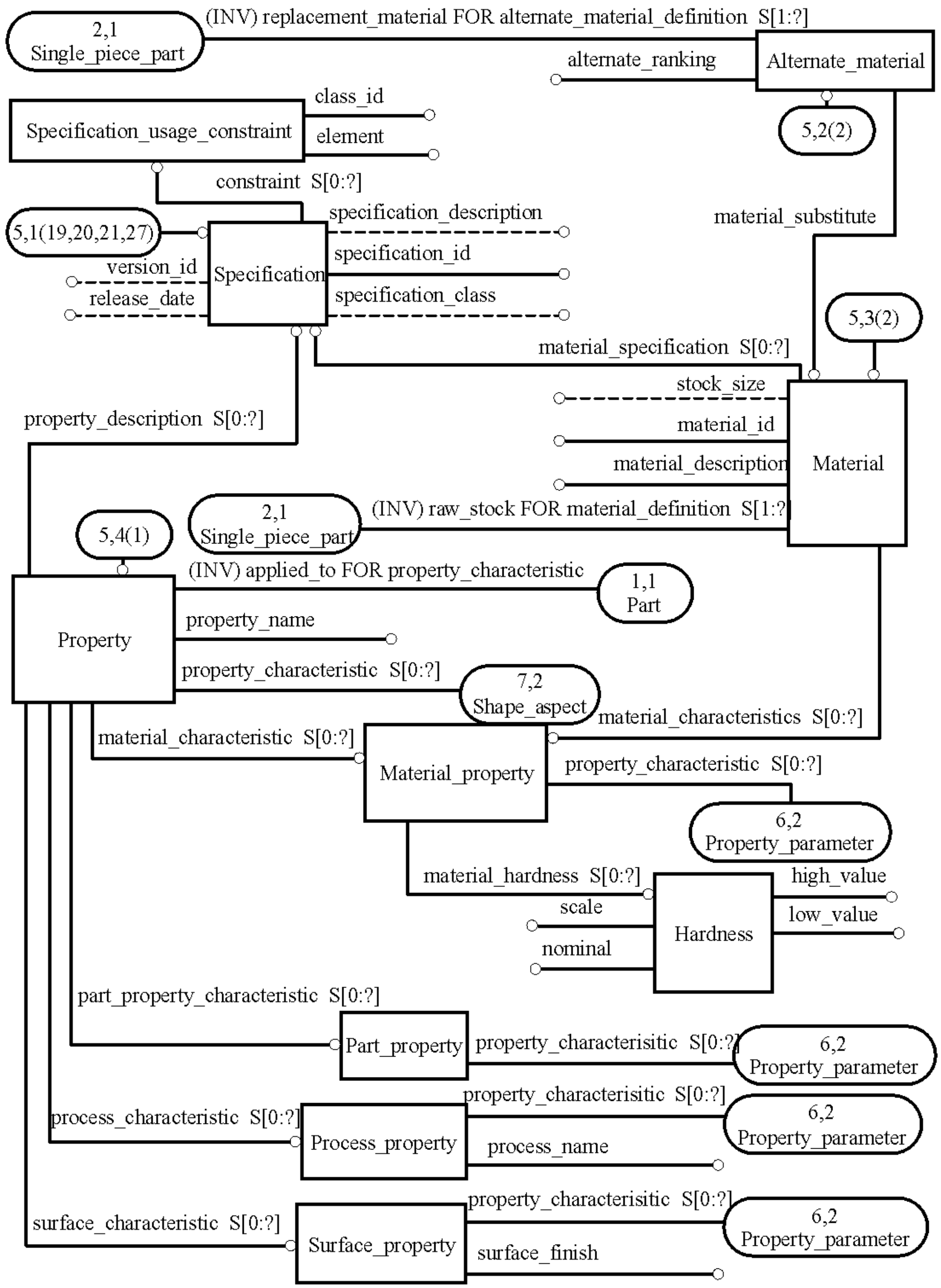


Figure G.5 — ARM EXPRESS-G diagram 5 of 30

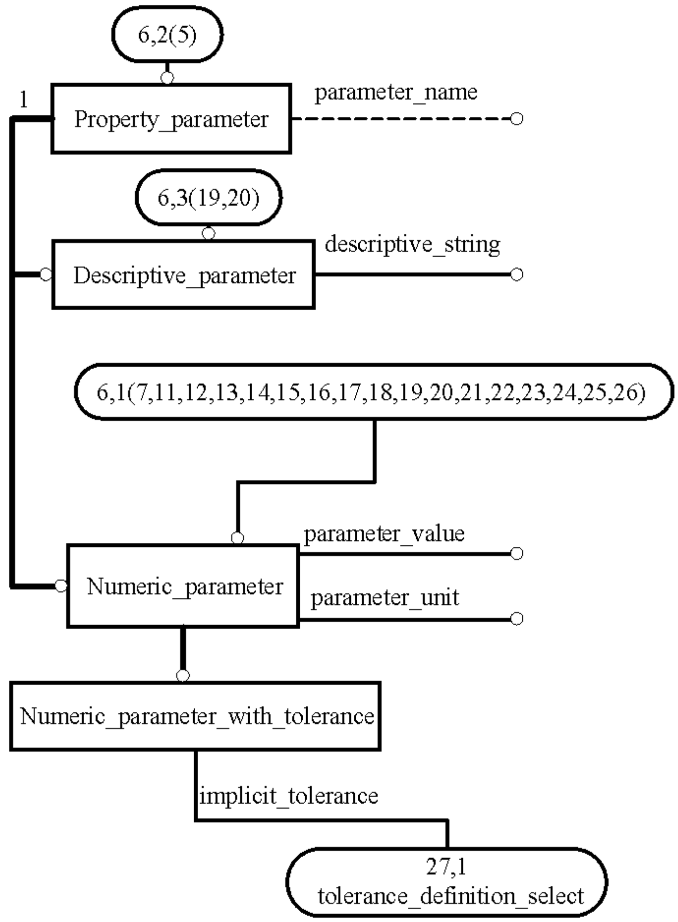


Figure G.6 — ARM EXPRESS-G diagram 6 of 30

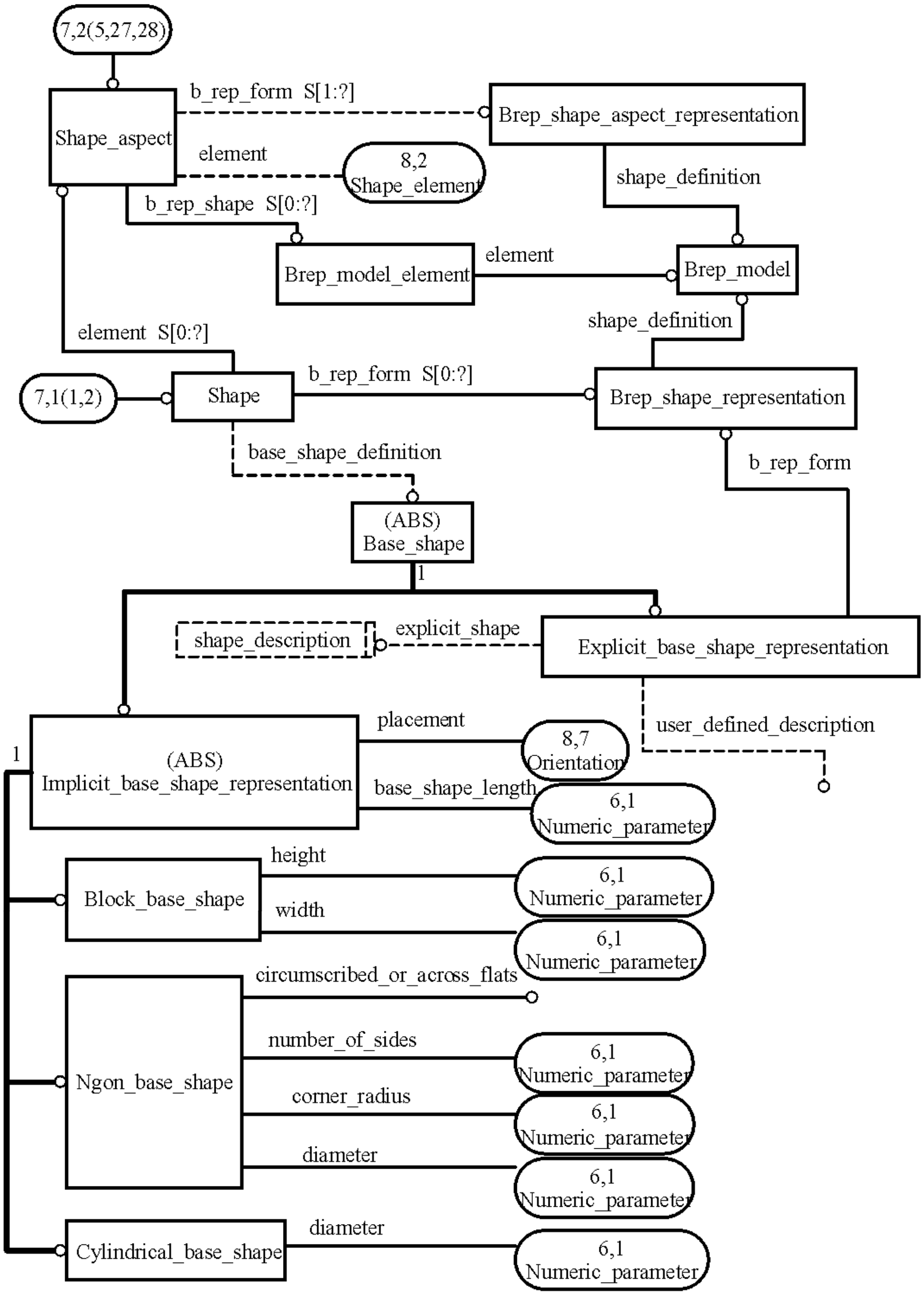


Figure G.7 — ARM EXPRESS-G diagram 7 of 30

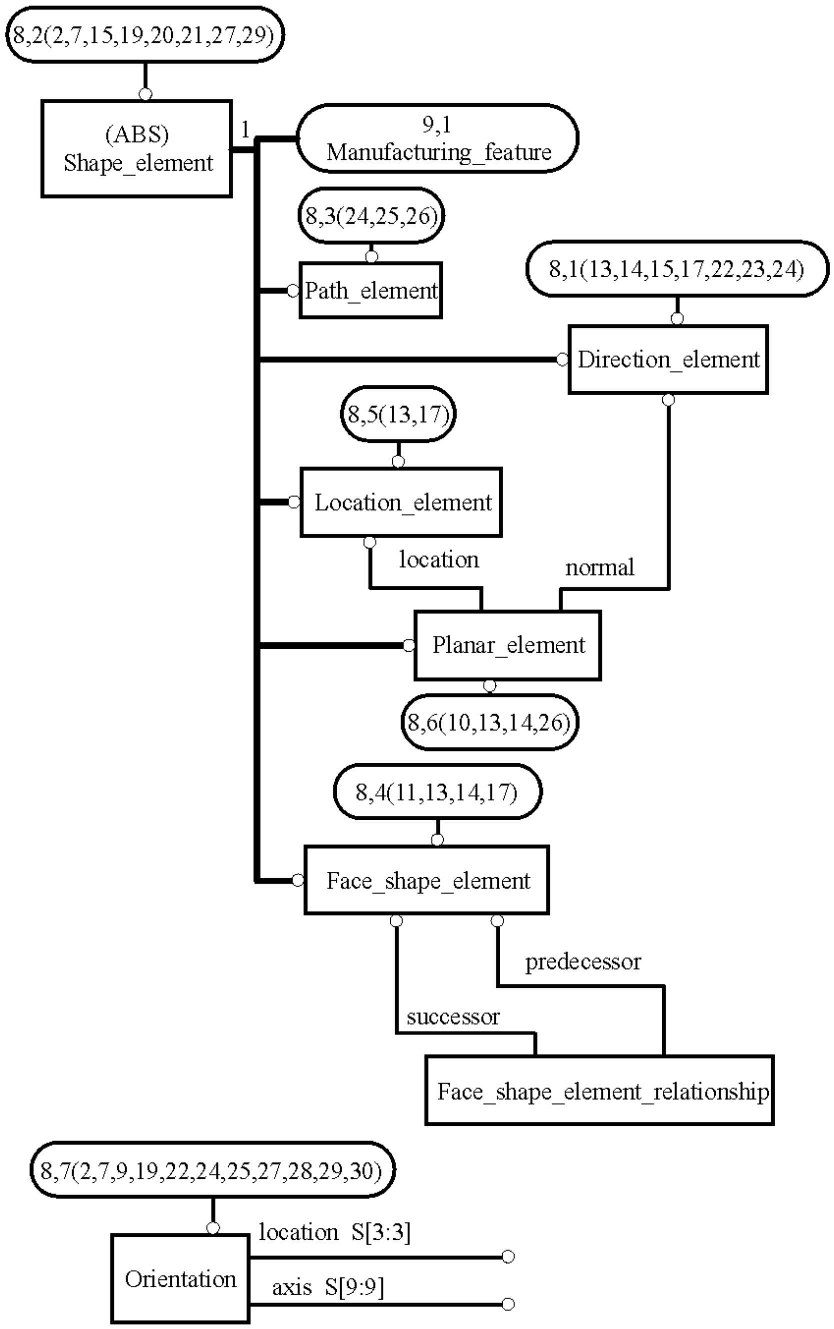


Figure G.8 — ARM EXPRESS-G diagram 8 of 30

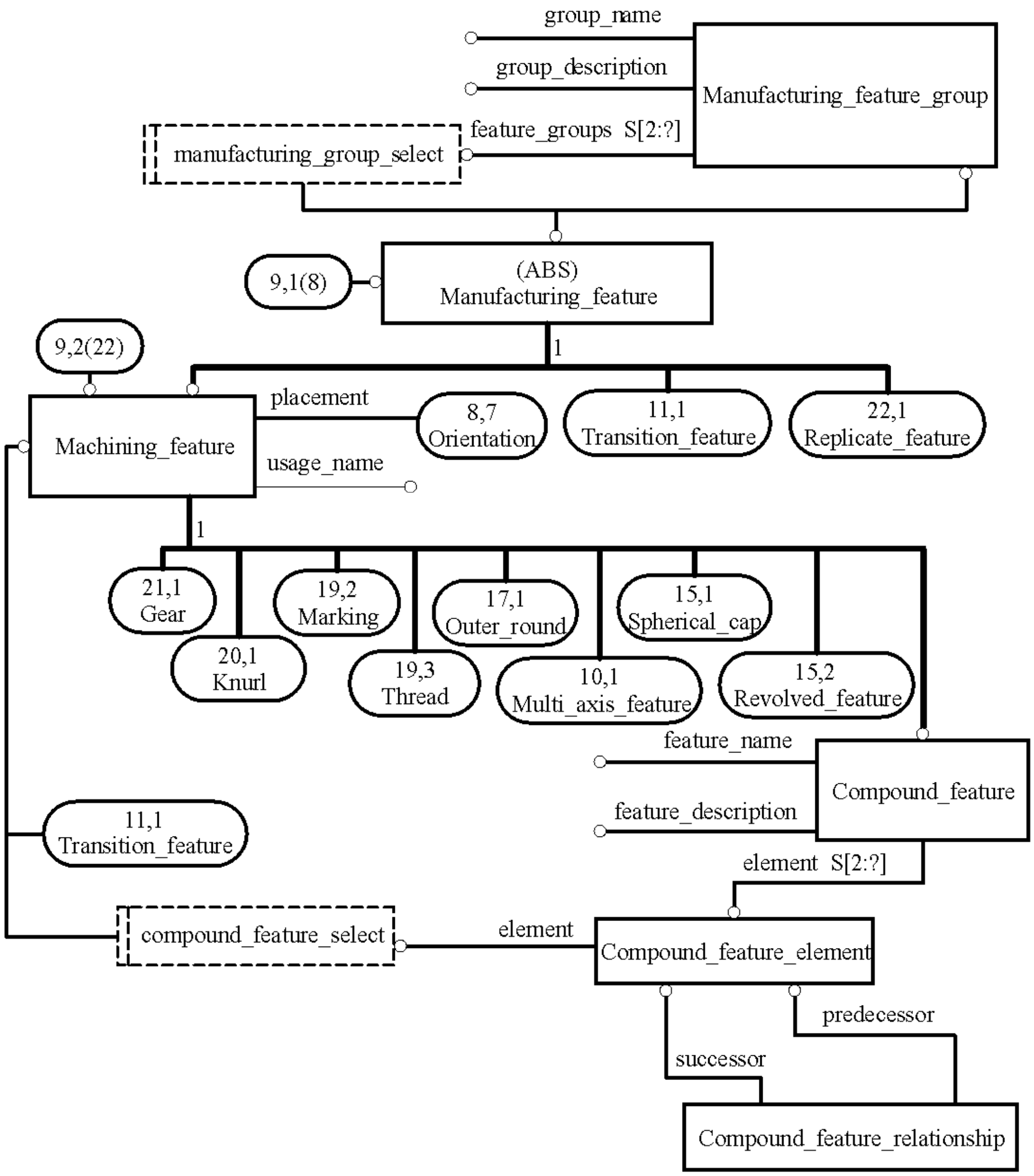


Figure G.9 — ARM EXPRESS-G diagram 9 of 30

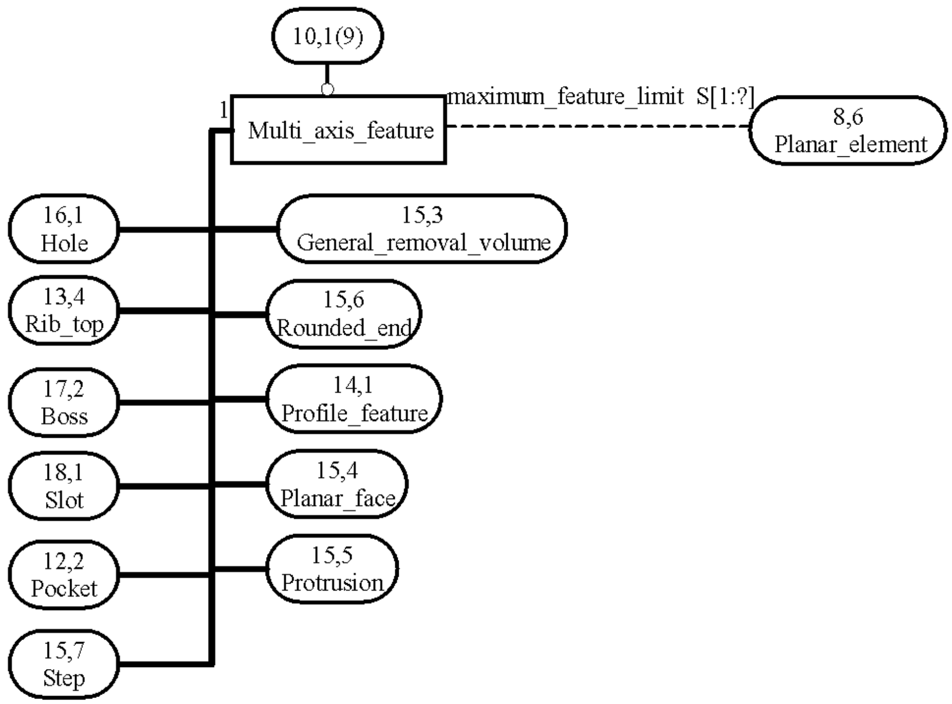


Figure G.10 — ARM EXPRESS-G diagram 10 of 30

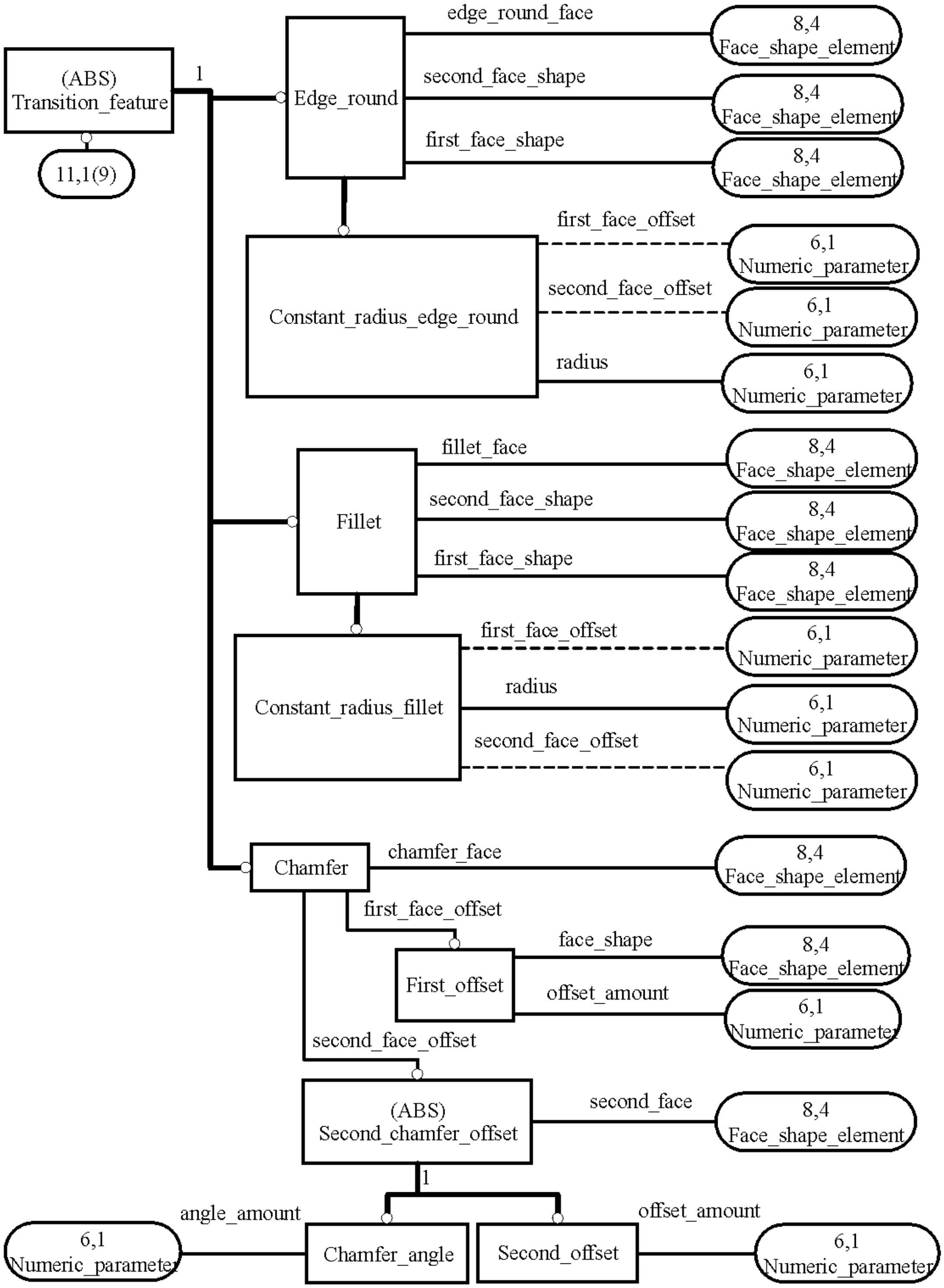


Figure G.11— ARM EXPRESS-G diagram 11 of 30

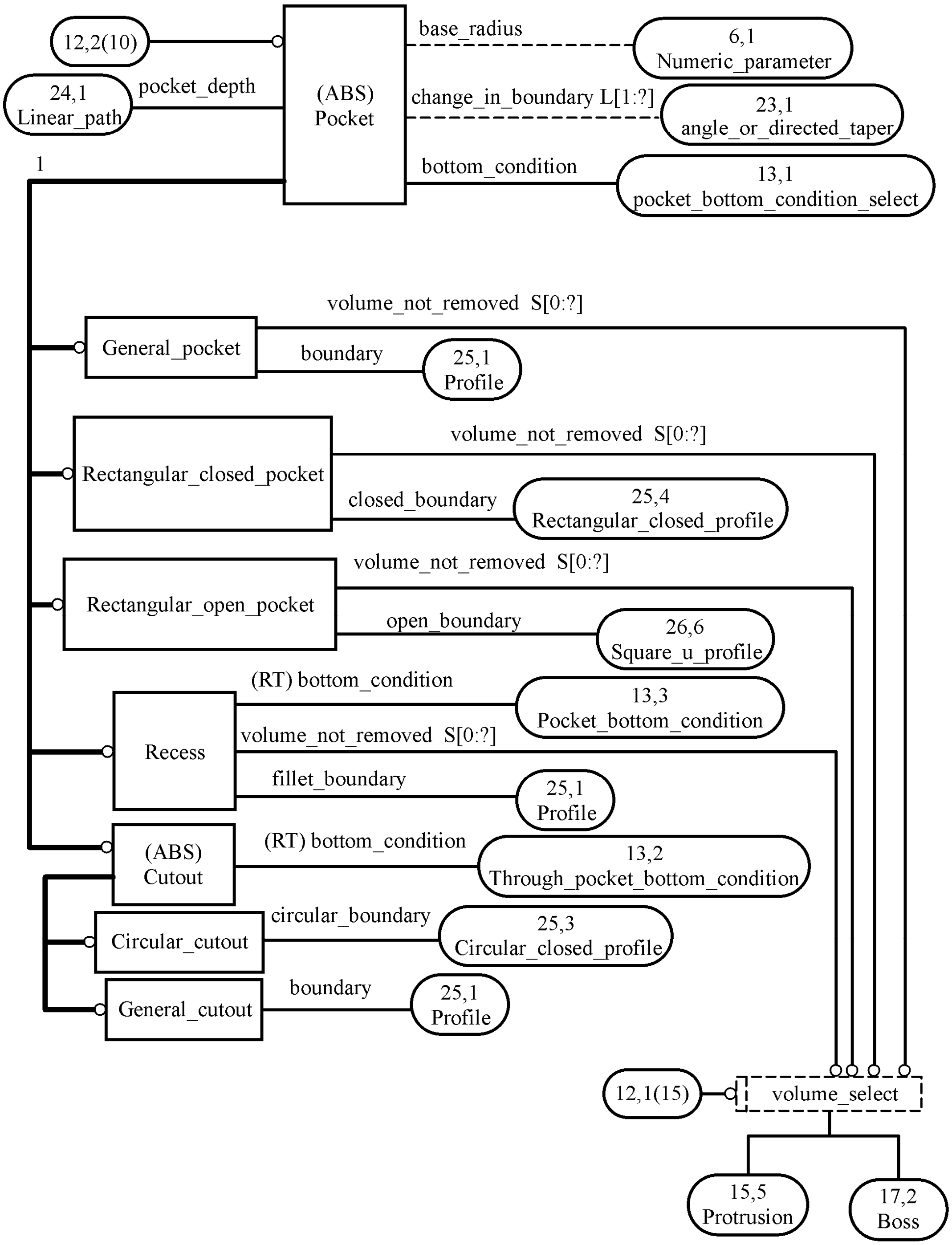


Figure G.12 — ARM EXPRESS-G diagram 12 of 30

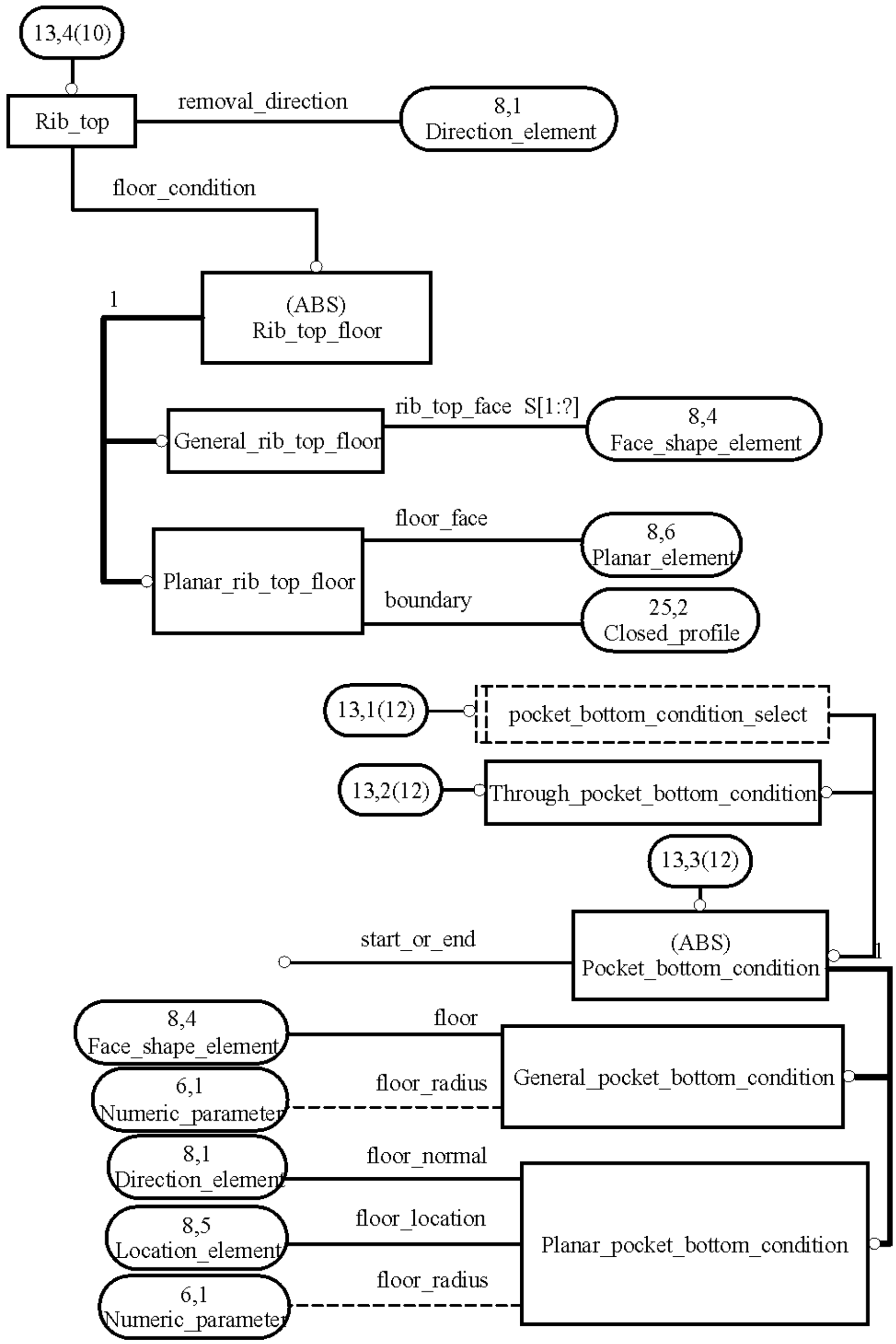


Figure G.13 — ARM EXPRESS-G diagram 13 of 30

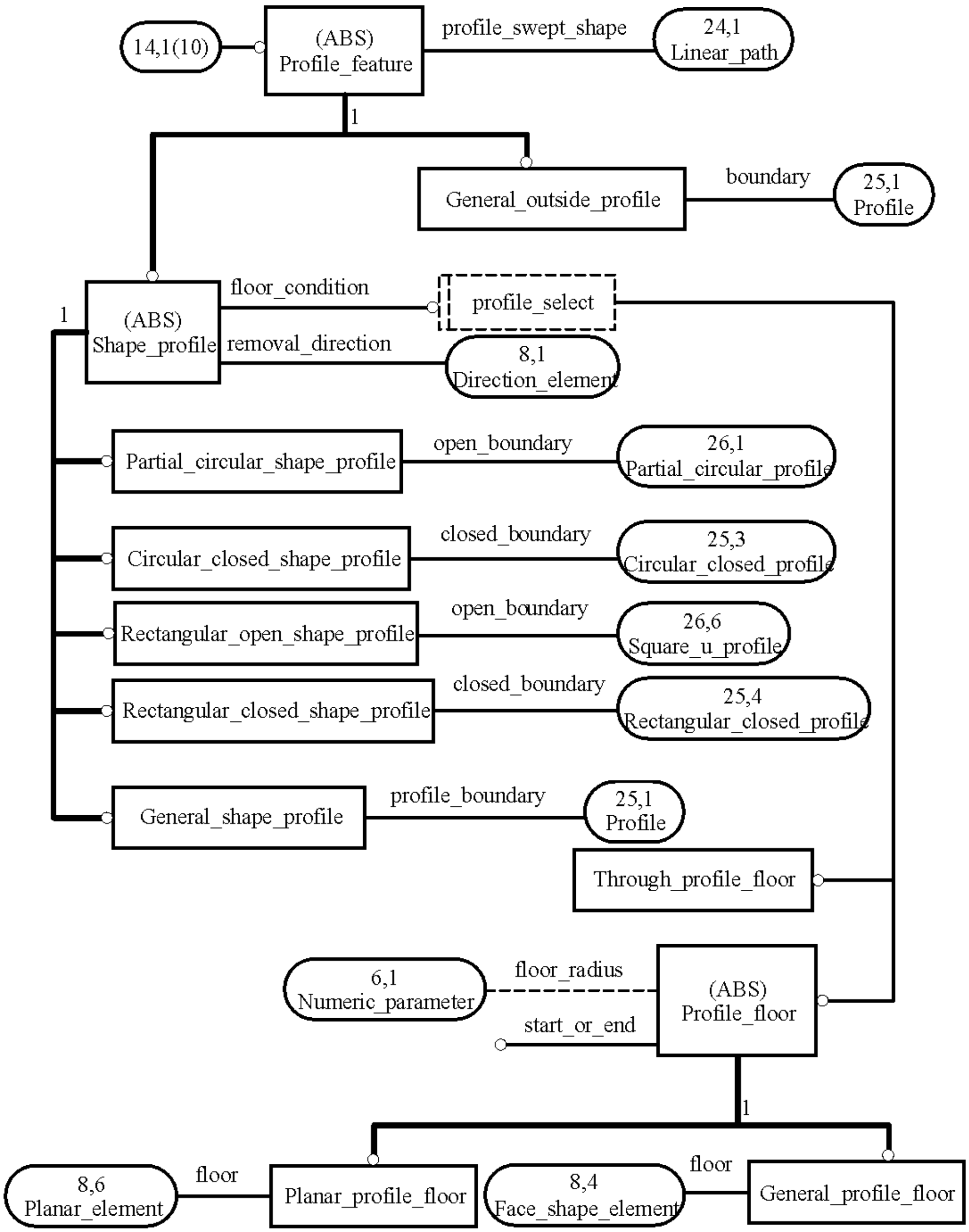


Figure G.14 — ARM EXPRESS-G diagram 14 of 30

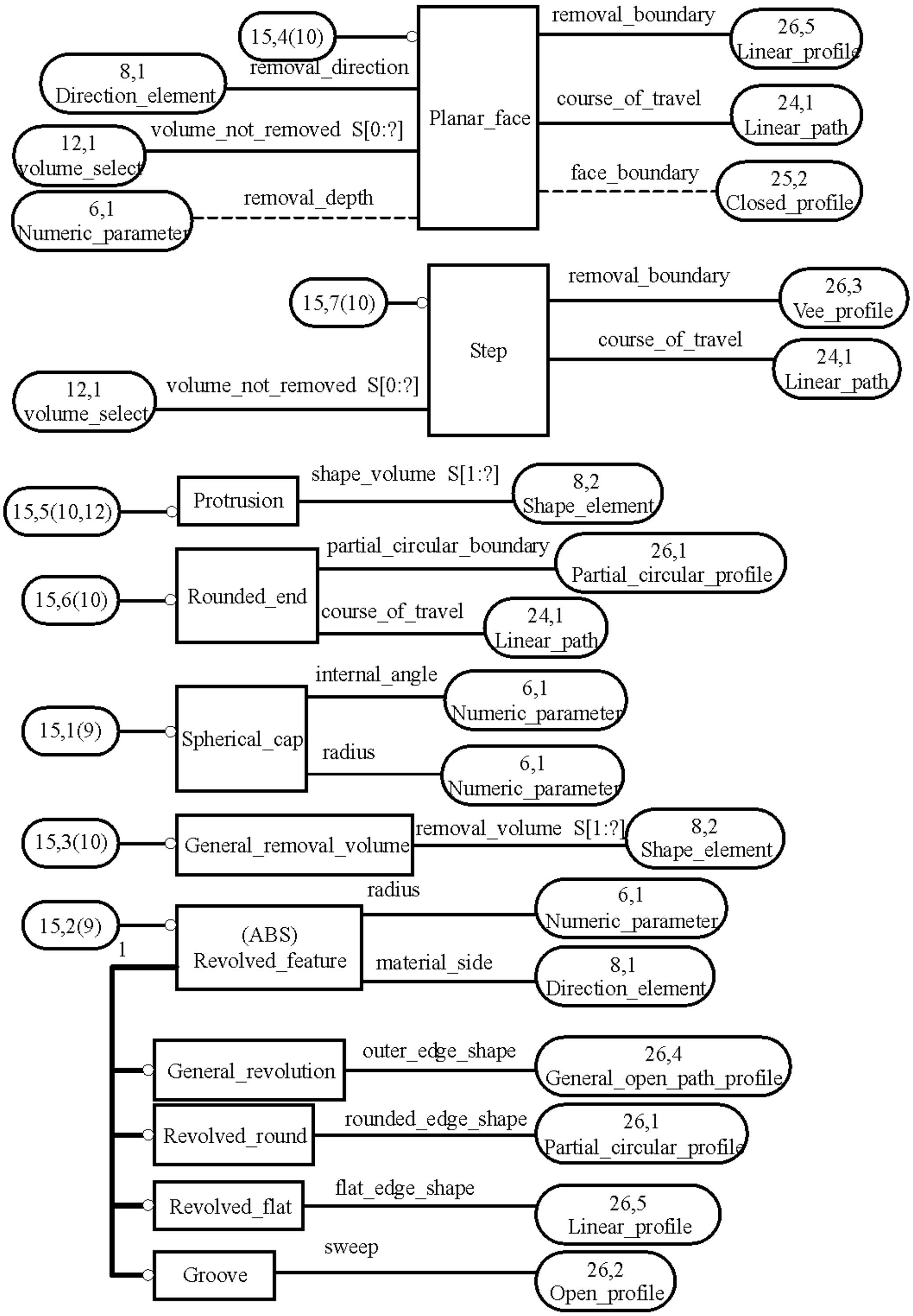


Figure G.15 — ARM EXPRESS-G diagram 15 of 30

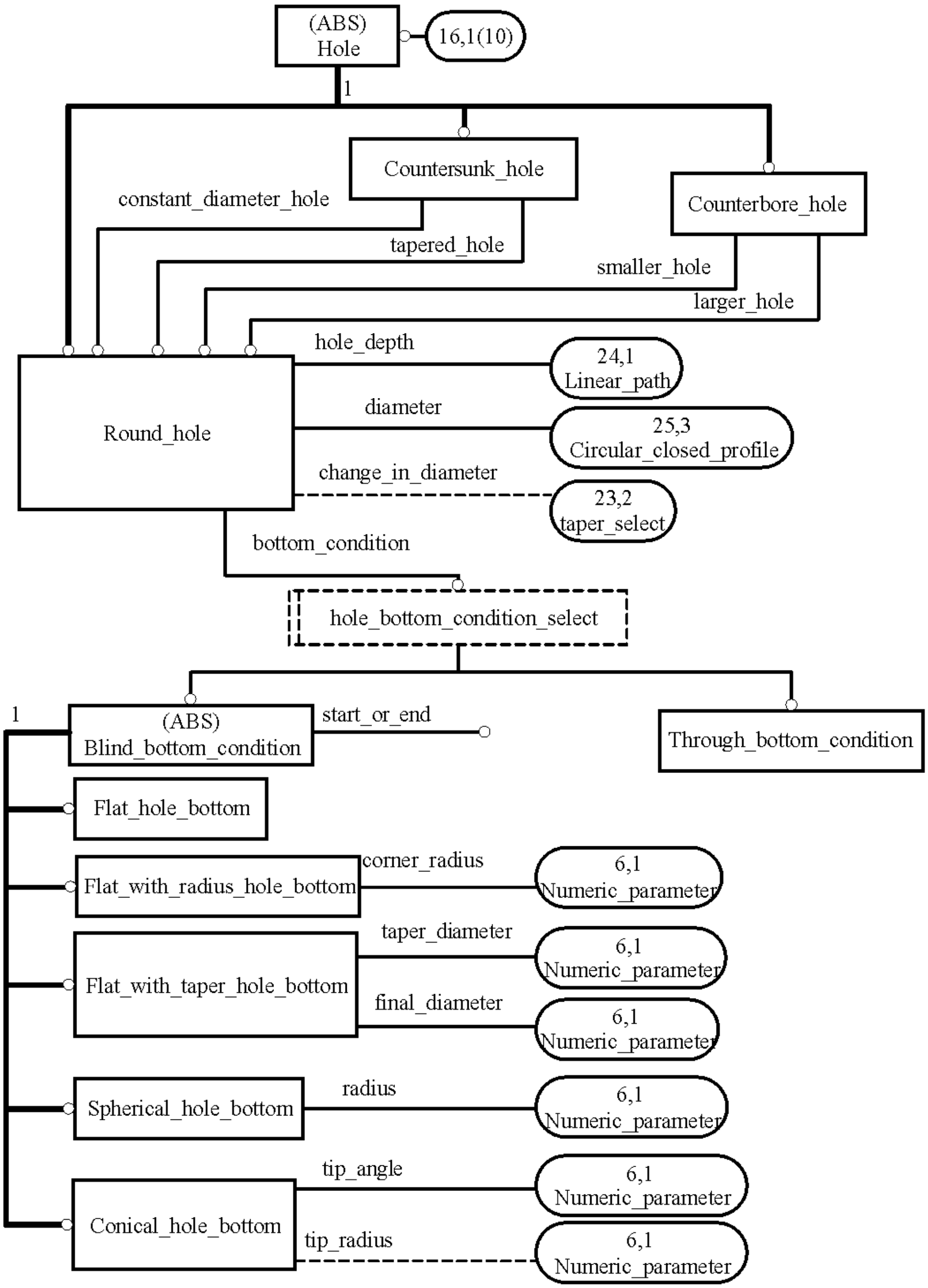


Figure G.16 — ARM EXPRESS-G diagram 16 of 30

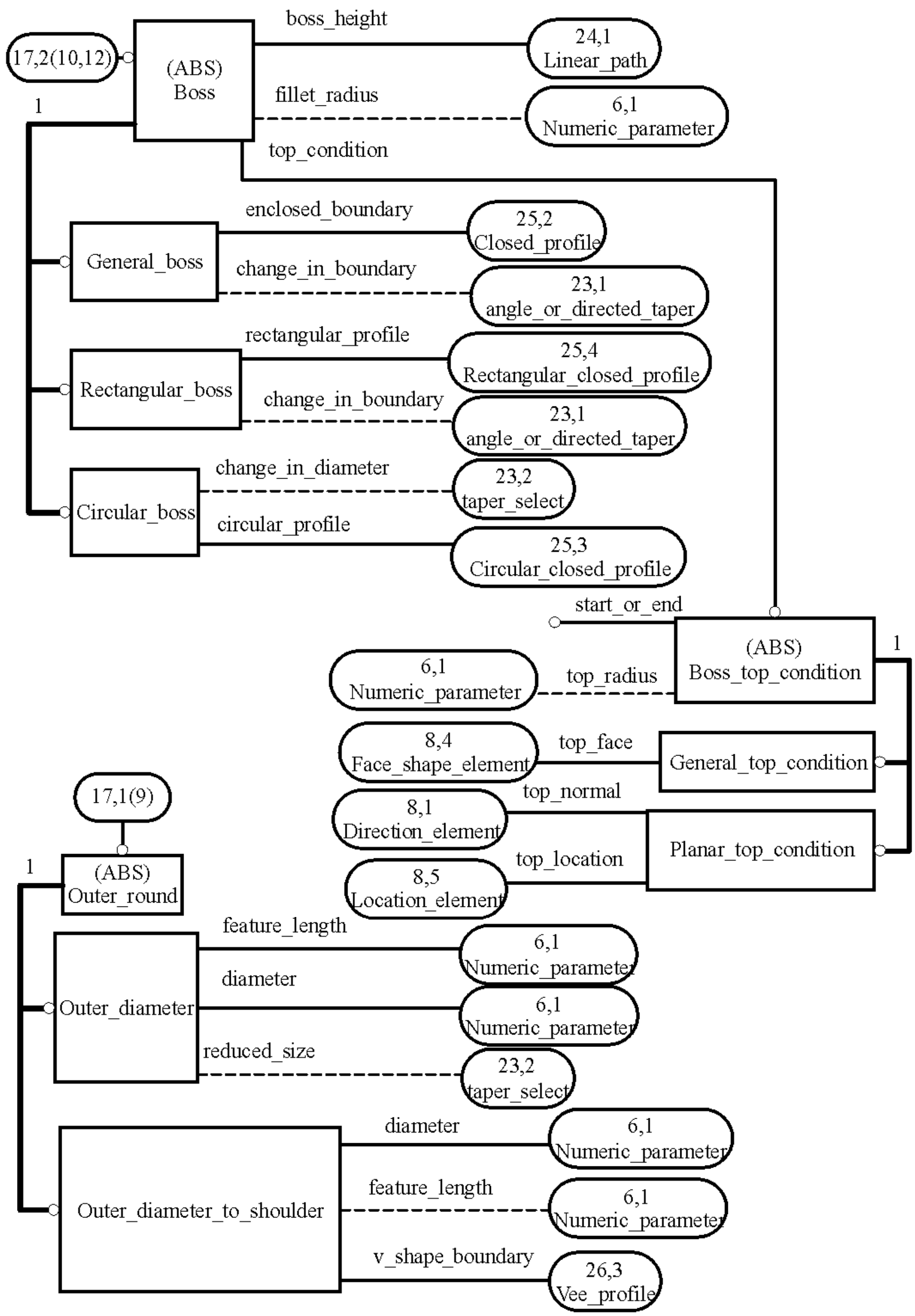


Figure G.17— ARM EXPRESS-G diagram 17 of 30

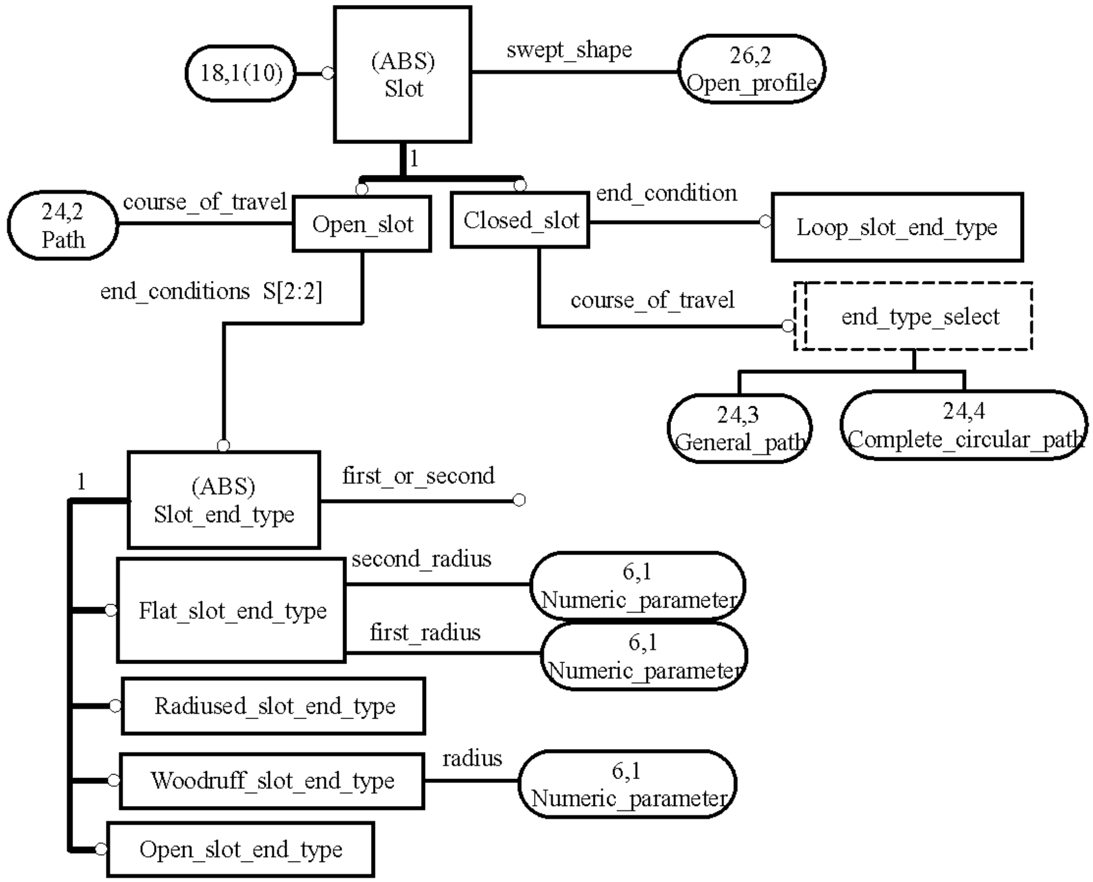


Figure G.18 — ARM EXPRESS-G diagram 18 of 30

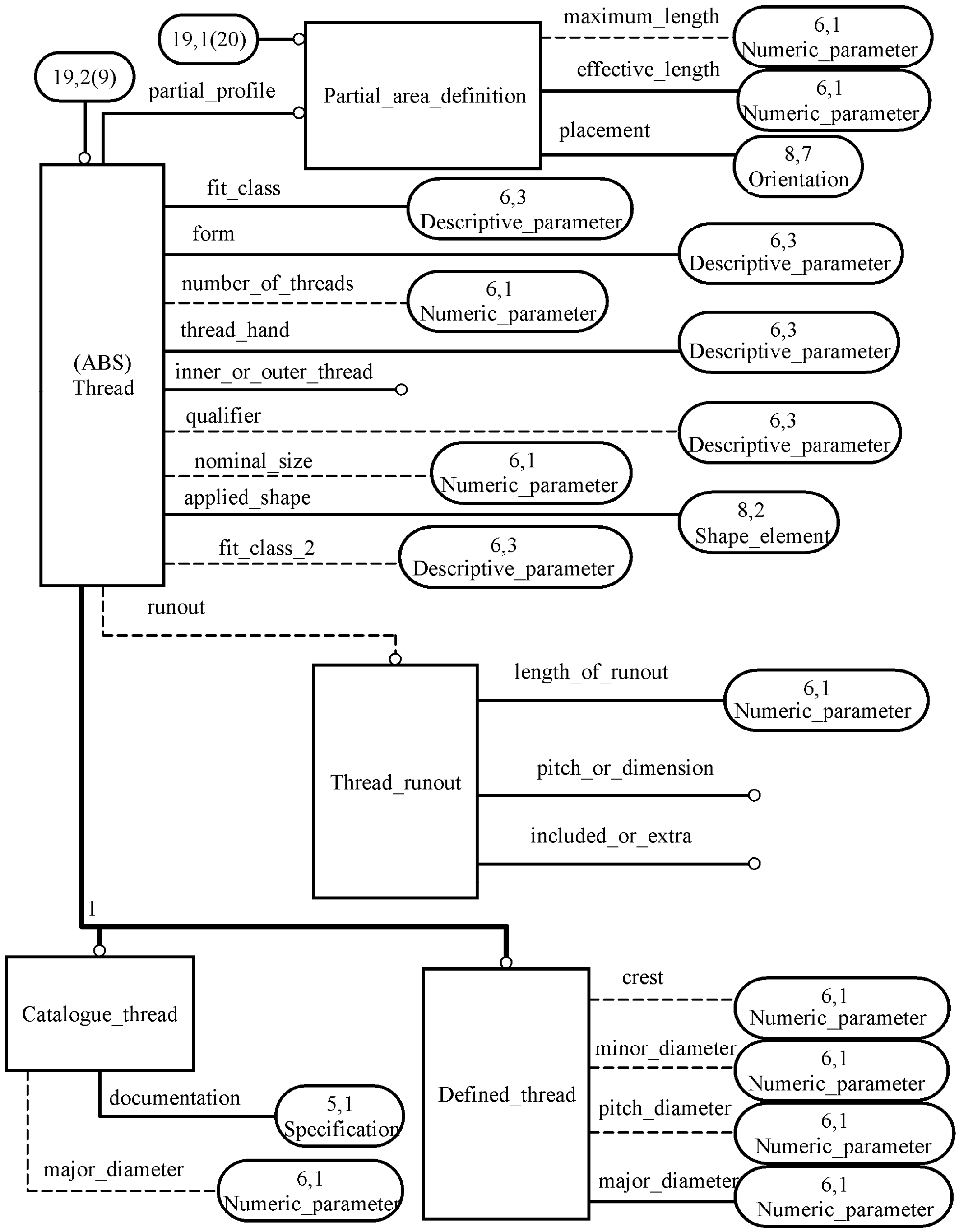


Figure G.19 — ARM EXPRESS-G diagram 19 of 30

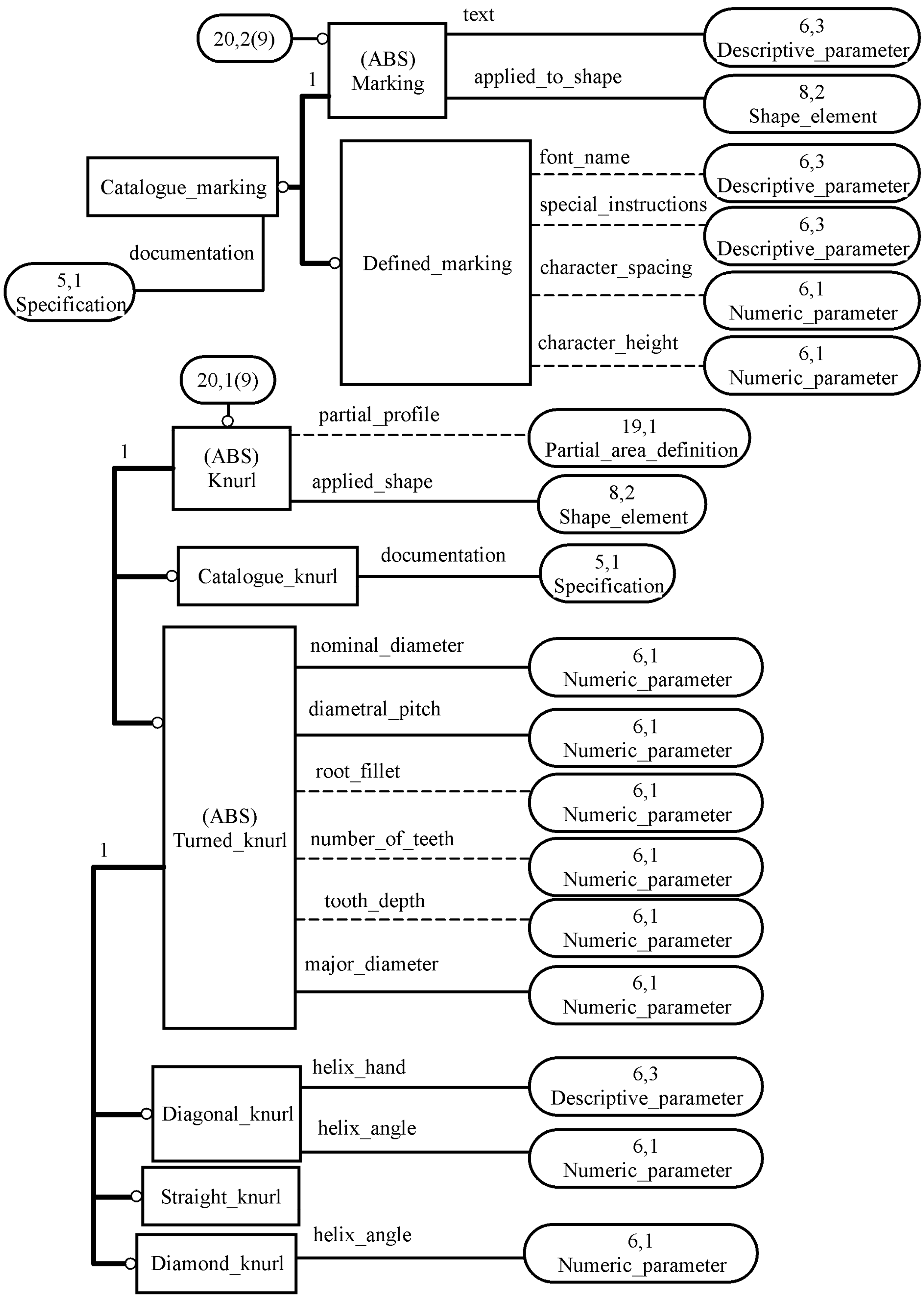


Figure G.20 — ARM EXPRESS-G diagram 20 of 30

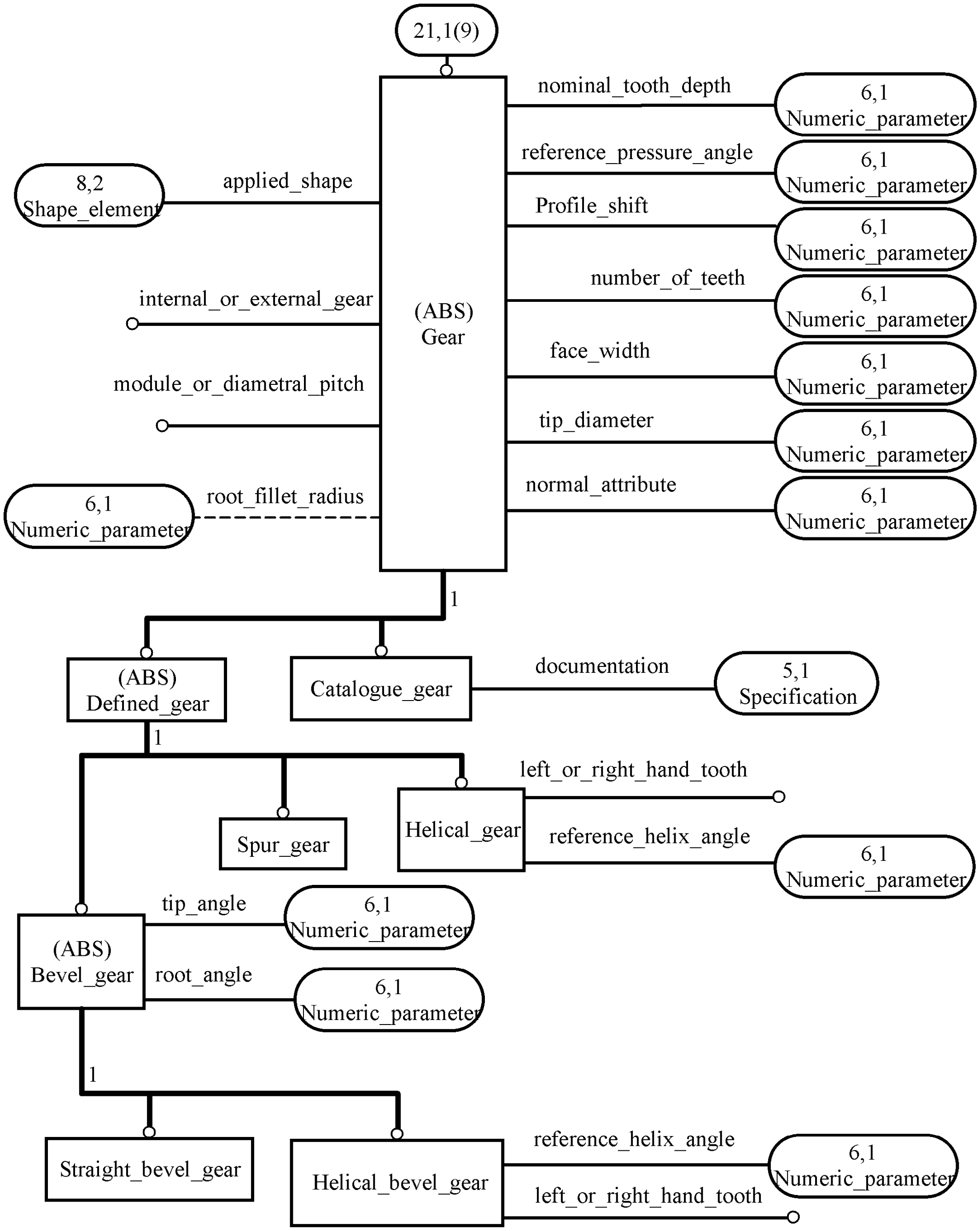


Figure G.21 — ARM EXPRESS-G diagram 21 of 30

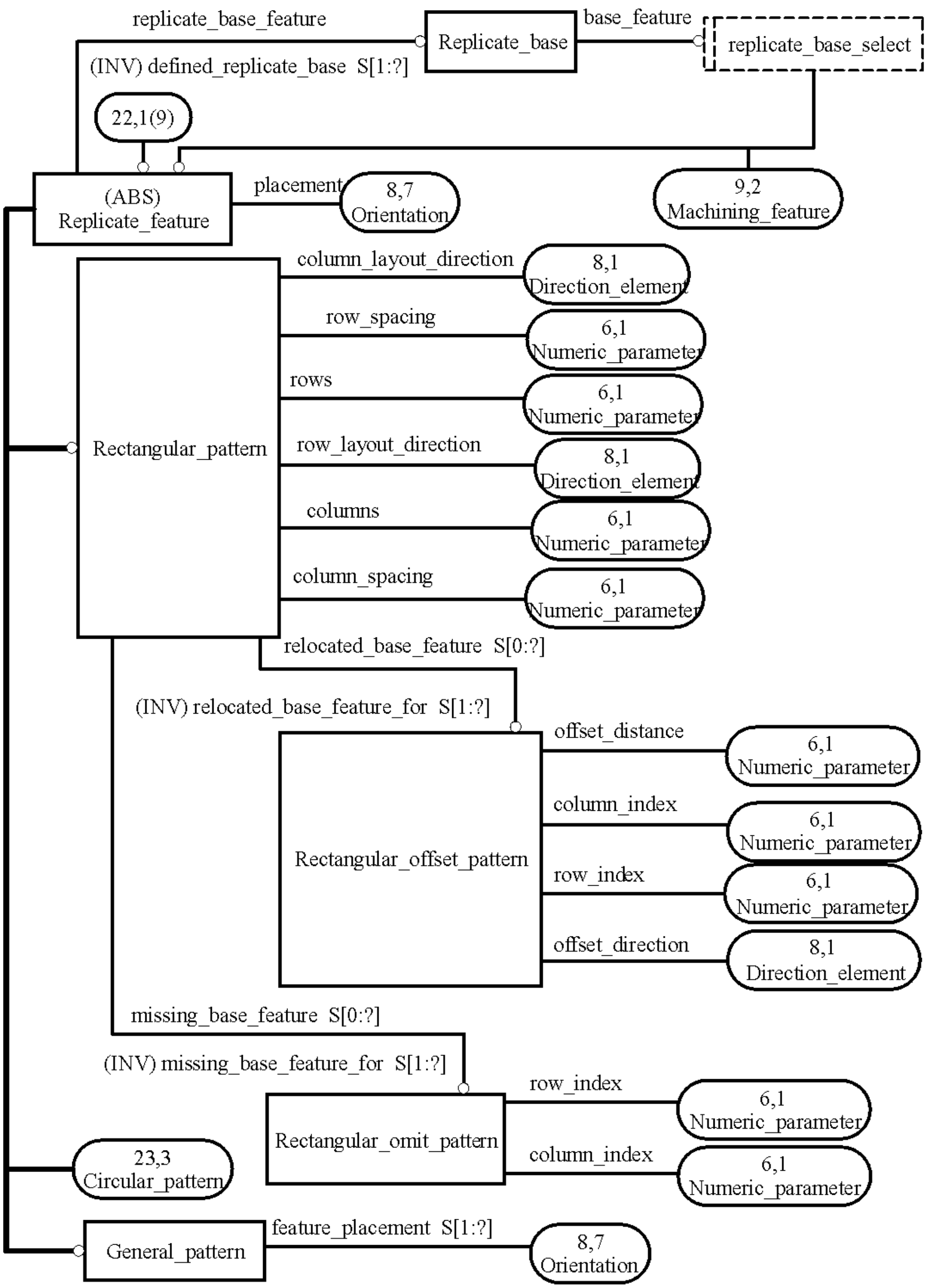


Figure G.22 — ARM EXPRESS-G diagram 22 of 30

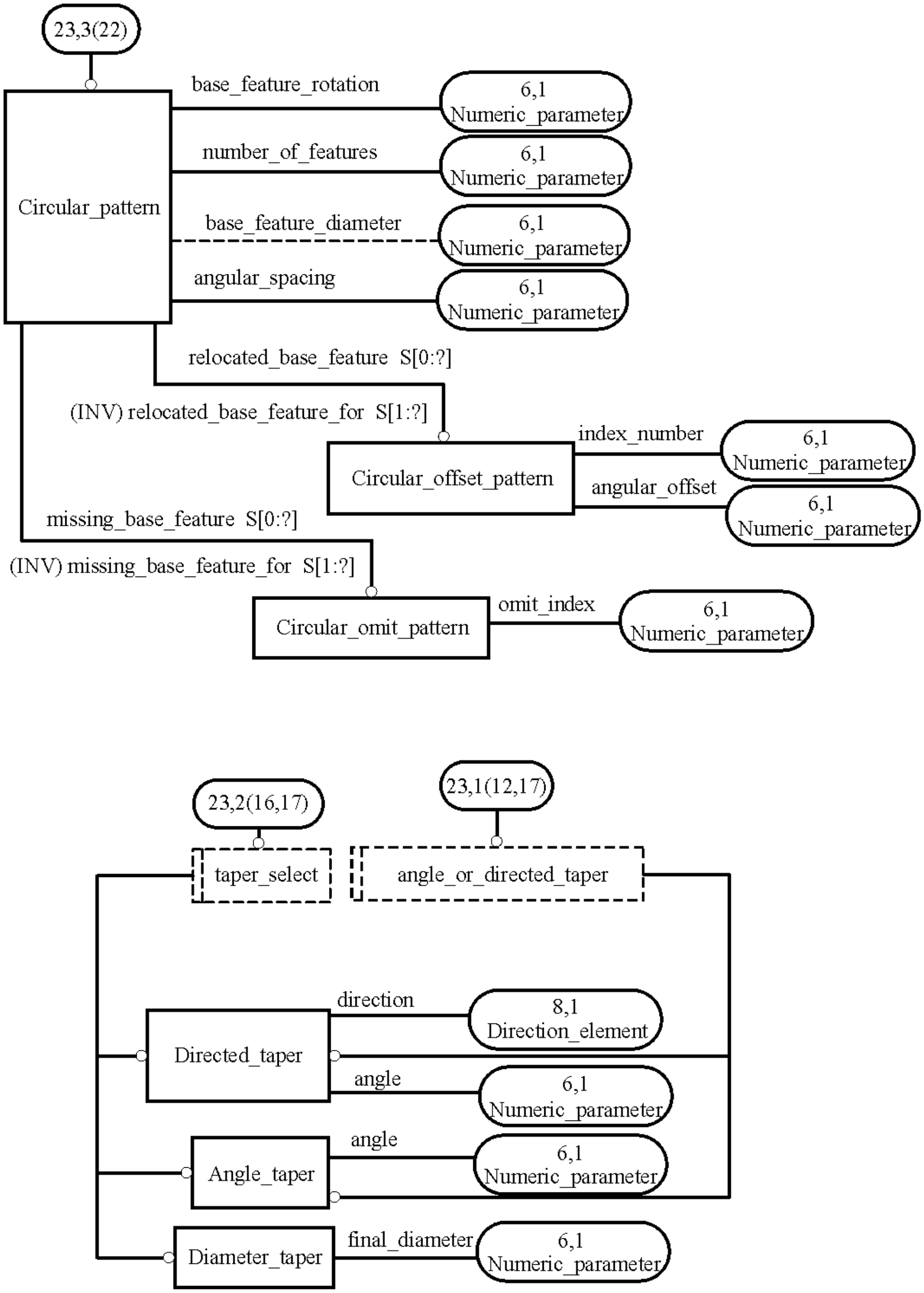


Figure G.23 — ARM EXPRESS-G diagram 23 of 30

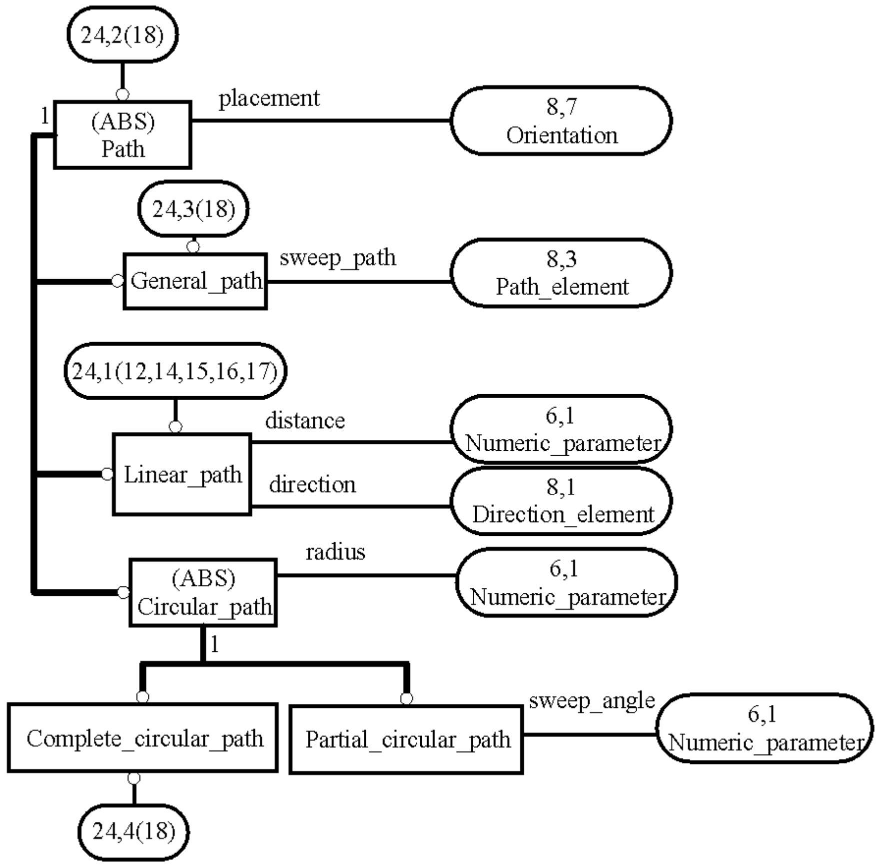


Figure G.24 — ARM EXPRESS-G diagram 24 of 30

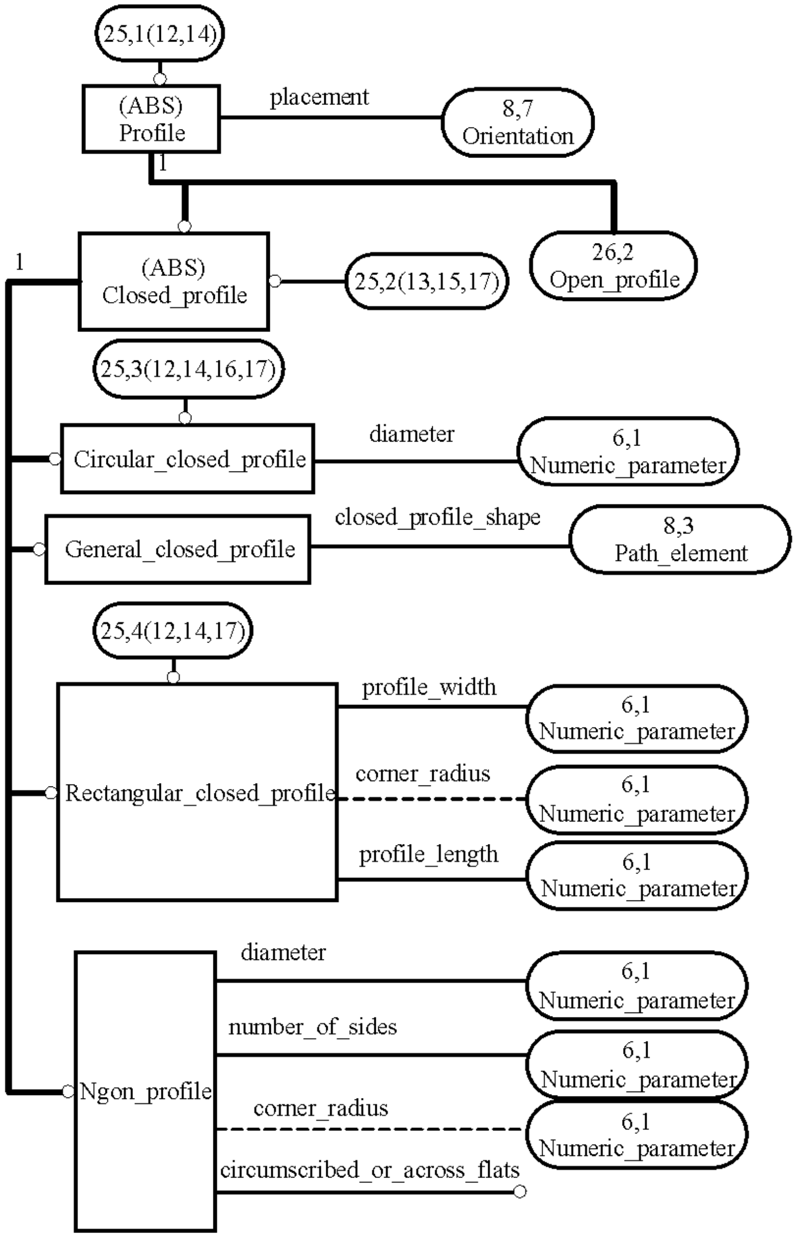


Figure G.25 — ARM EXPRESS-G diagram 25 of 30

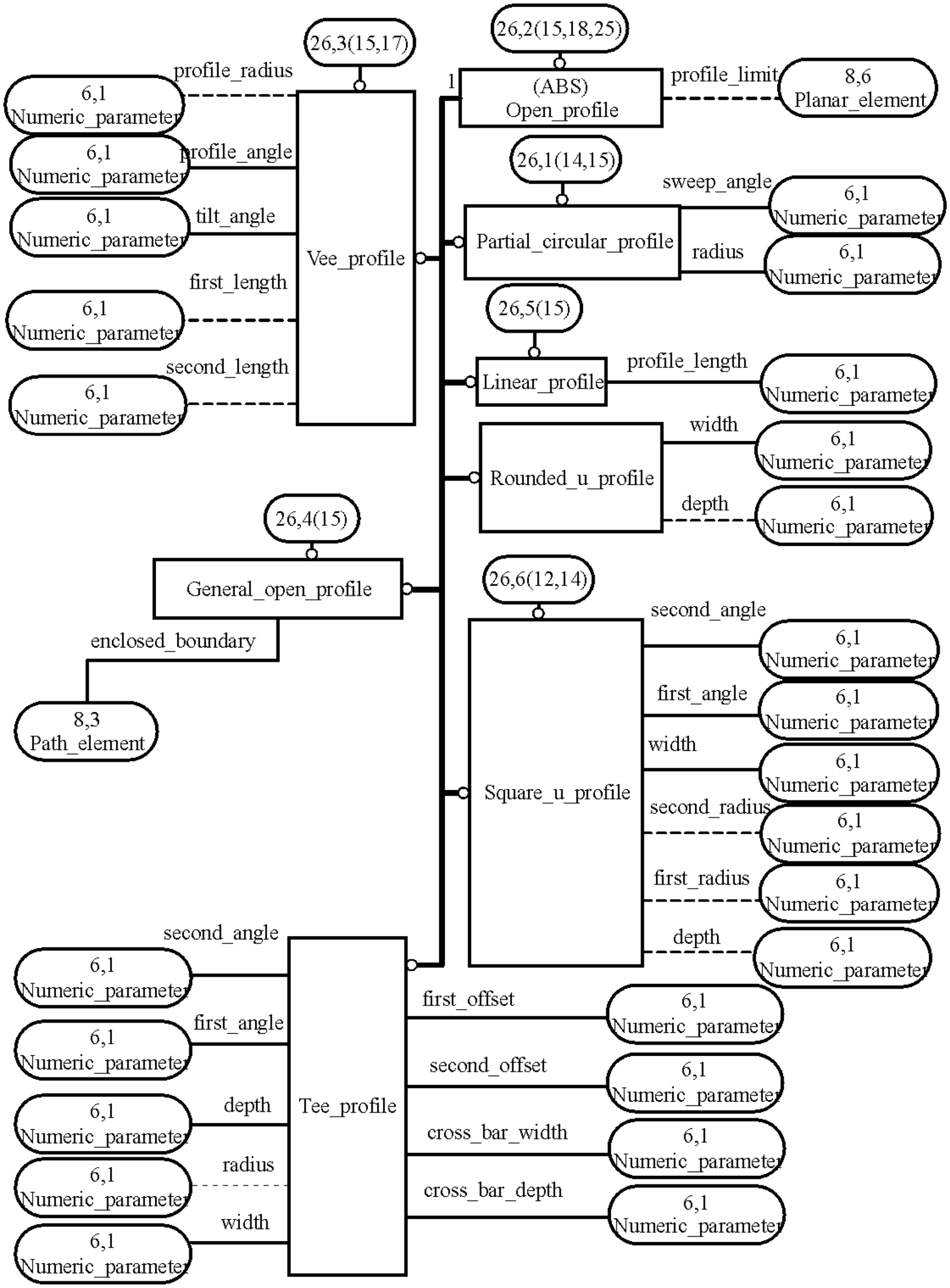


Figure G.26 — ARM EXPRESS-G diagram 26 of 30

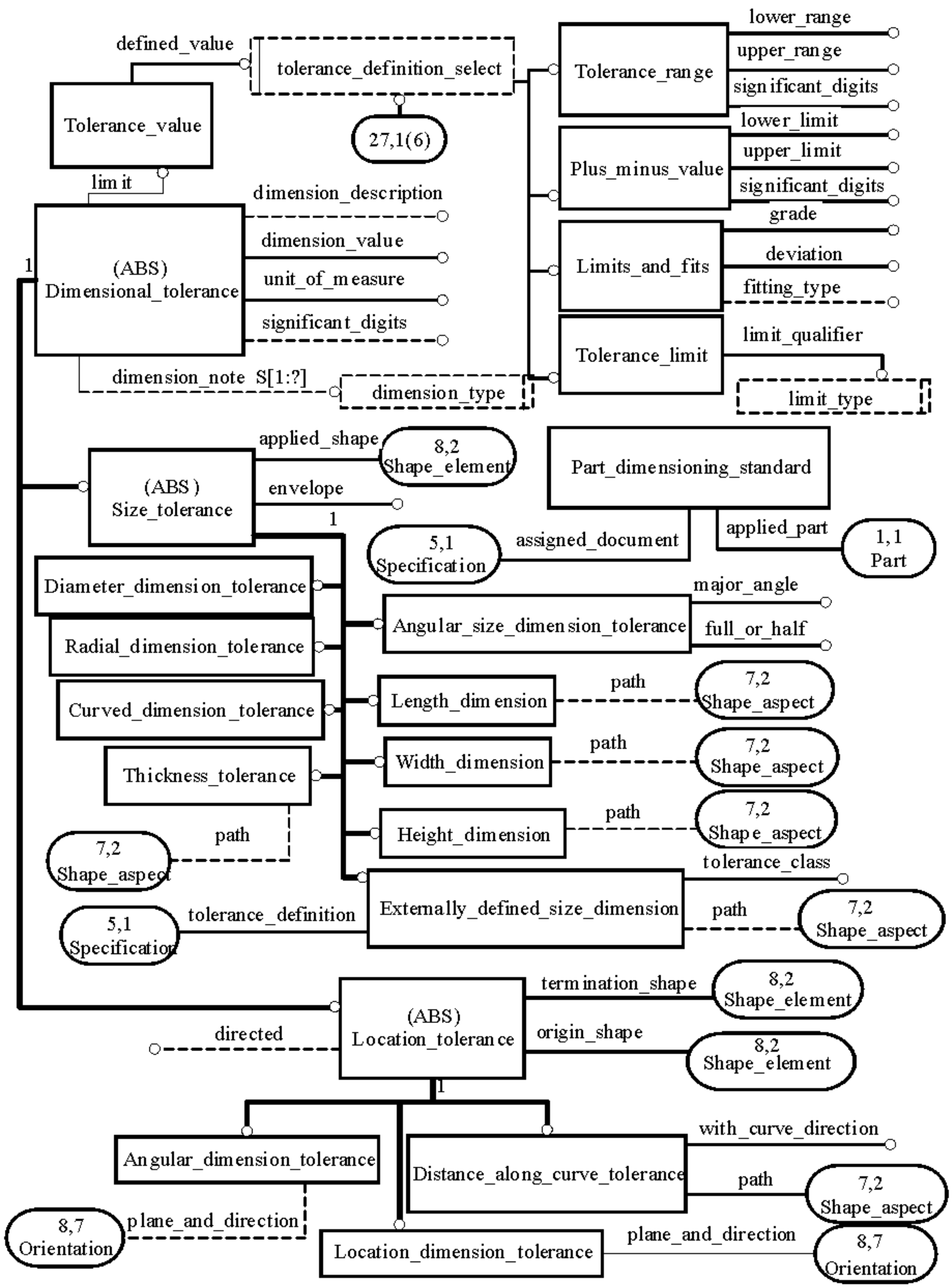


Figure G.27 — ARM EXPRESS-G diagram 27 of 30

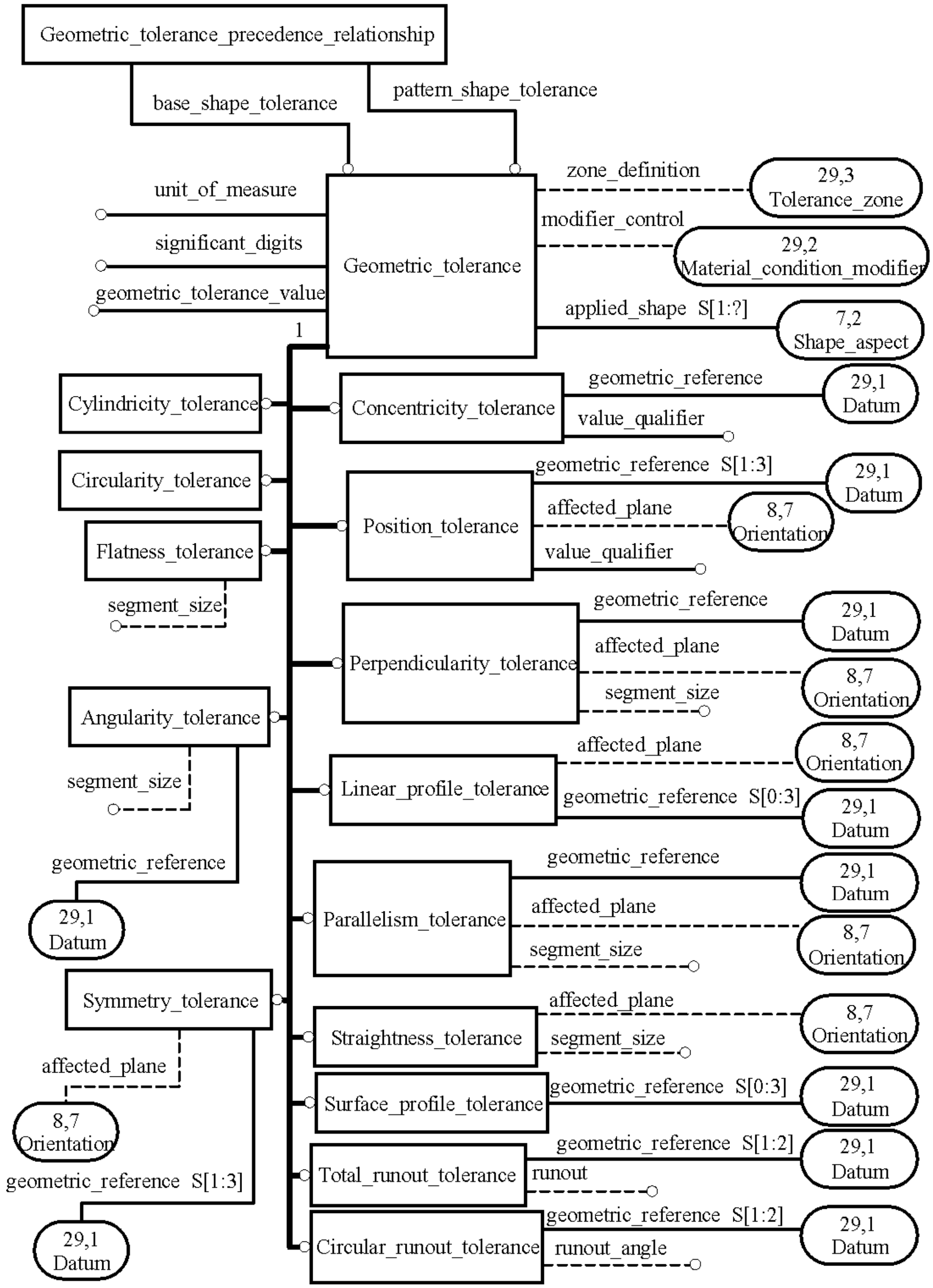


Figure G.28 — ARM EXPRESS-G diagram 28 of 30

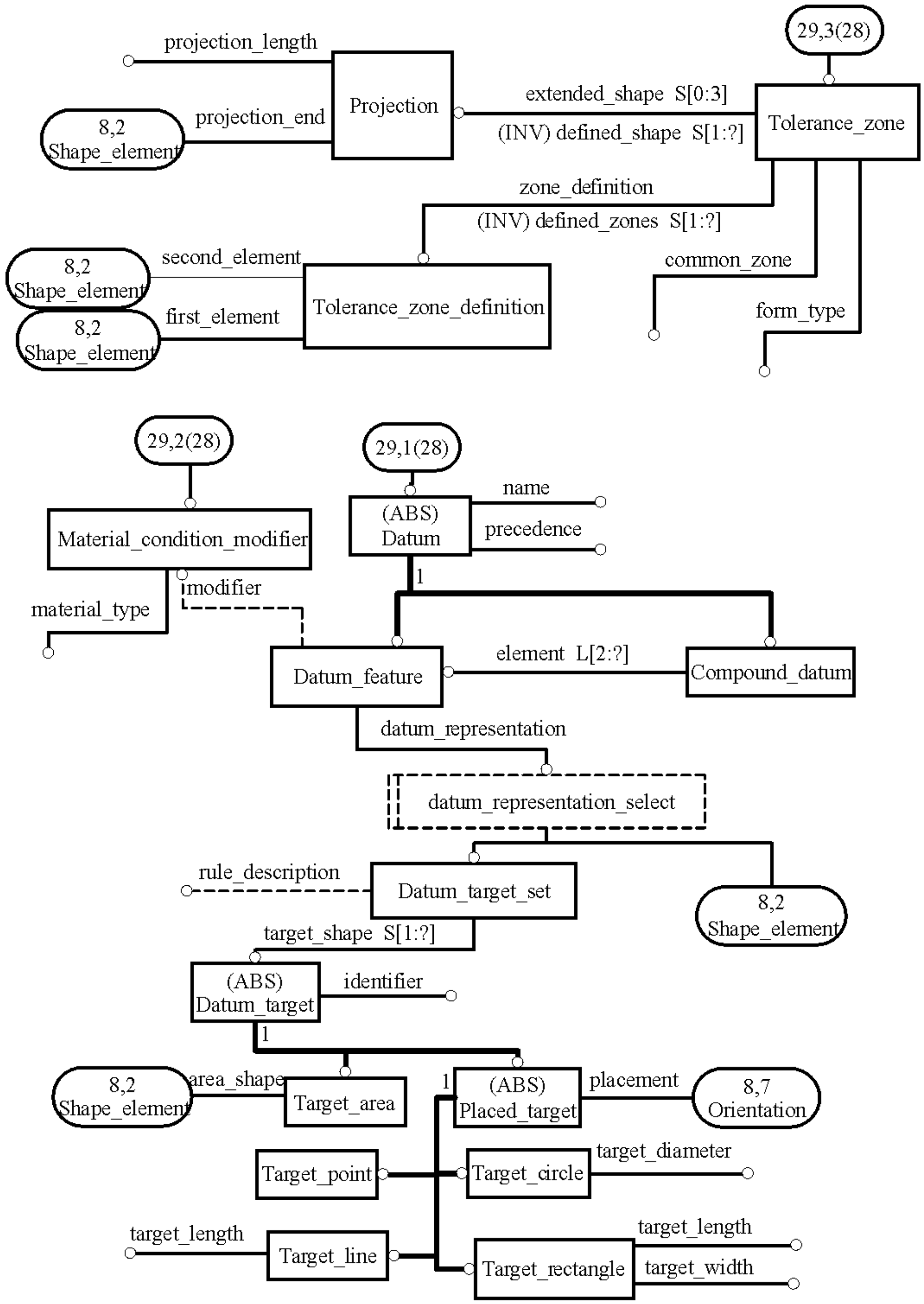


Figure G.29 — ARM EXPRESS-G diagram 29 of 30

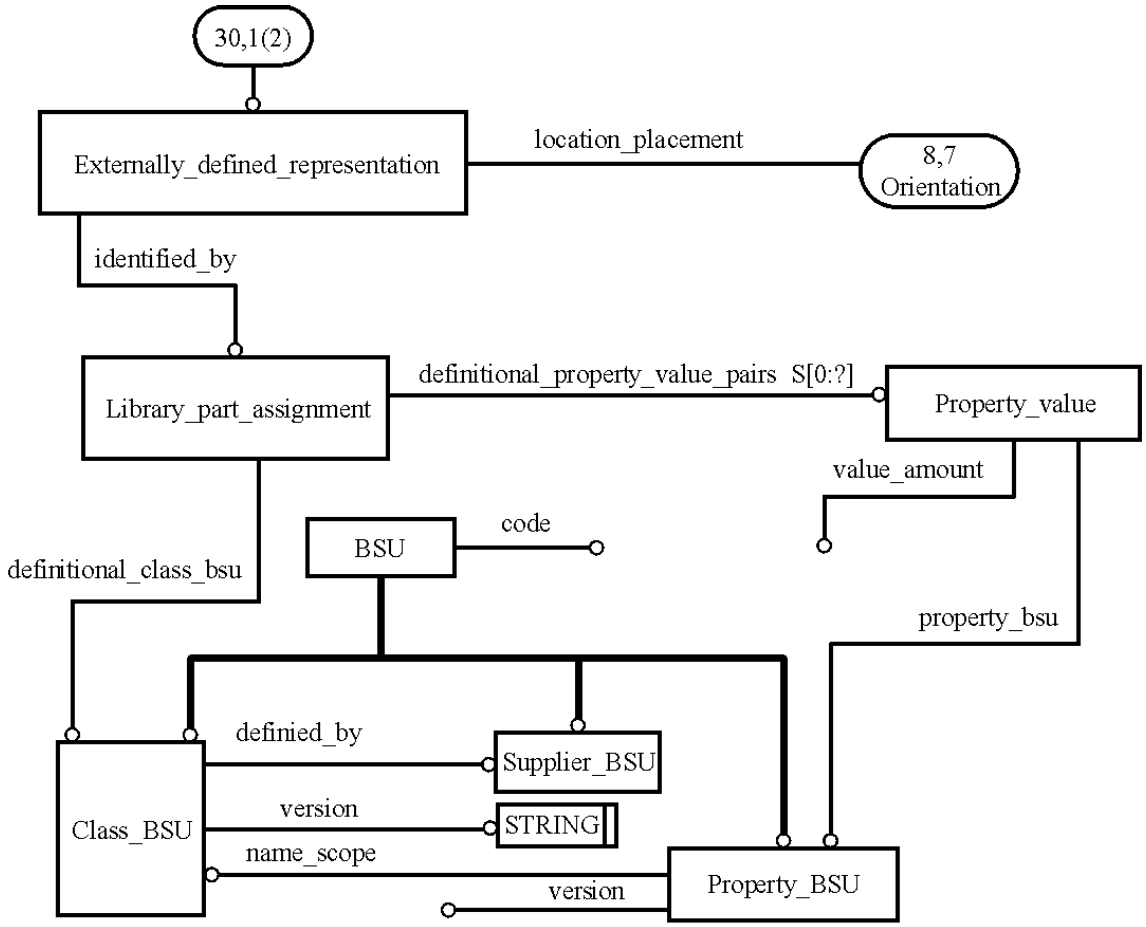


Figure G.30 — ARM EXPRESS-G diagram 30 of 30

Annex H
(informative)
AIM EXPRESS-G

Figures H.1 to H.30 correspond to the AIM EXPRESS annotated listing given in Annex A. The figures use the EXPRESS-G graphical notation for the EXPRESS language. EXPRESS-G is defined in Annex A of ISO 10303-11.

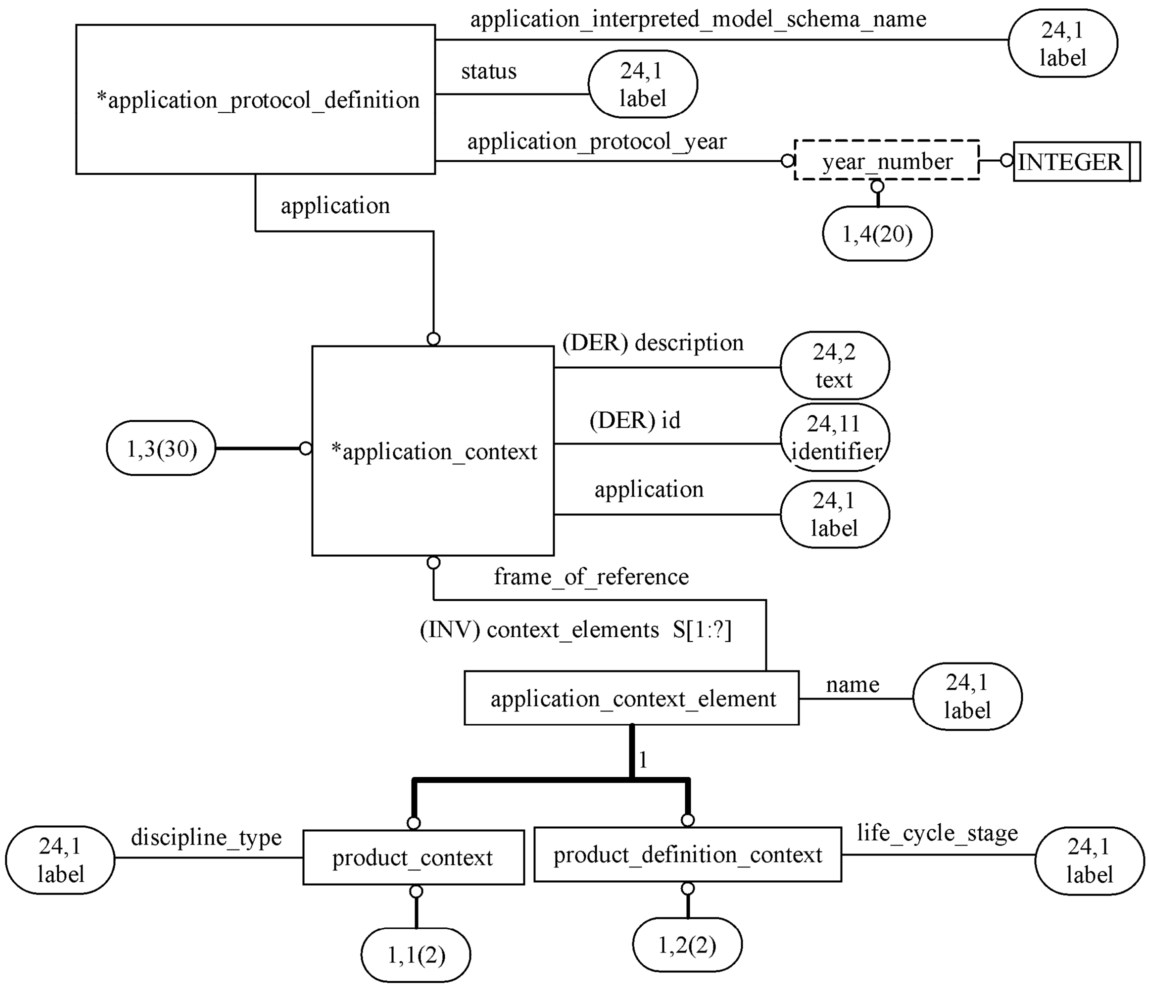


Figure H.1 — application context — AIM EXPRESS-G diagram 1 of 30

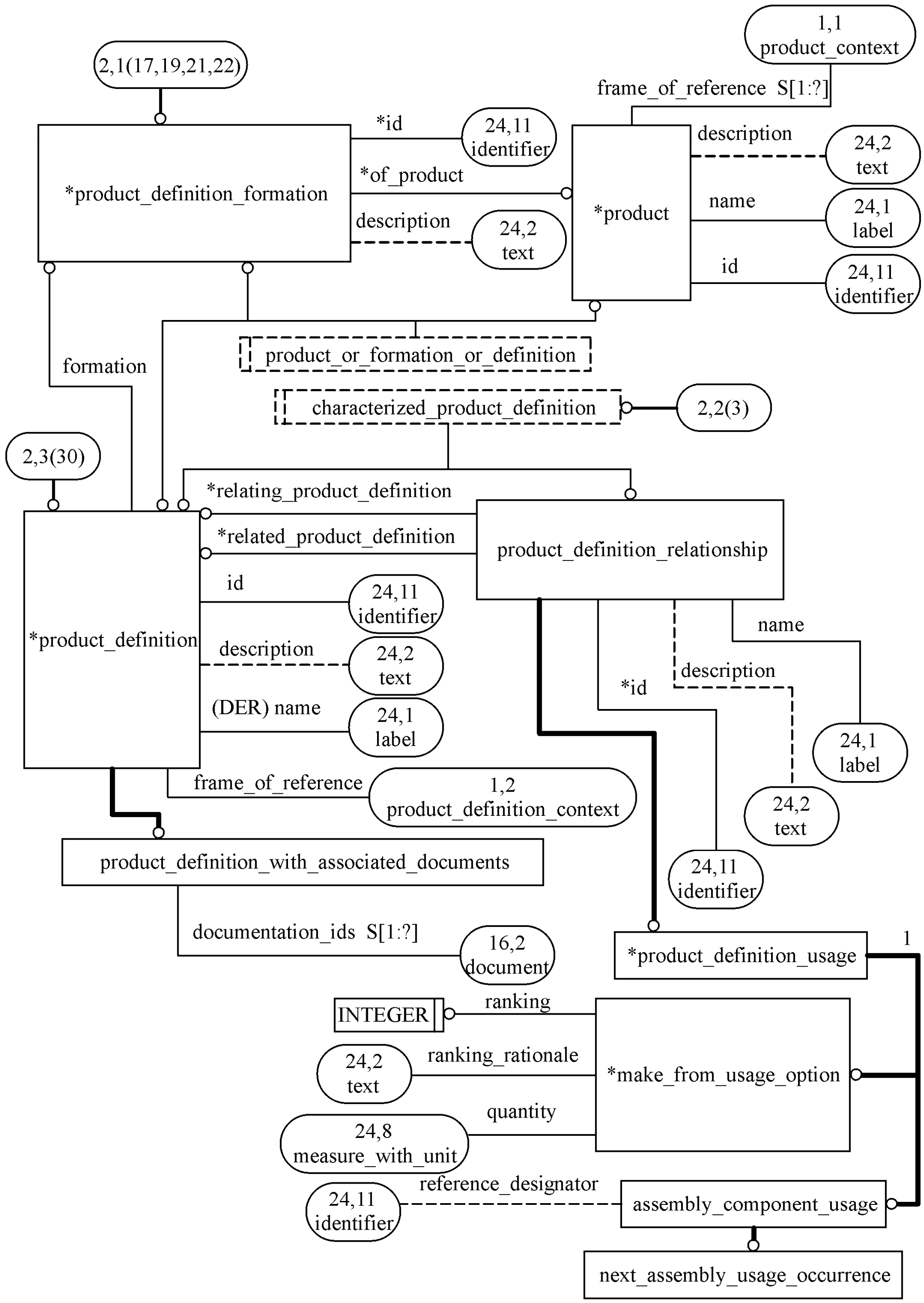


Figure H.2 — product definition — AIM EXPRESS-G diagram 2 of 30

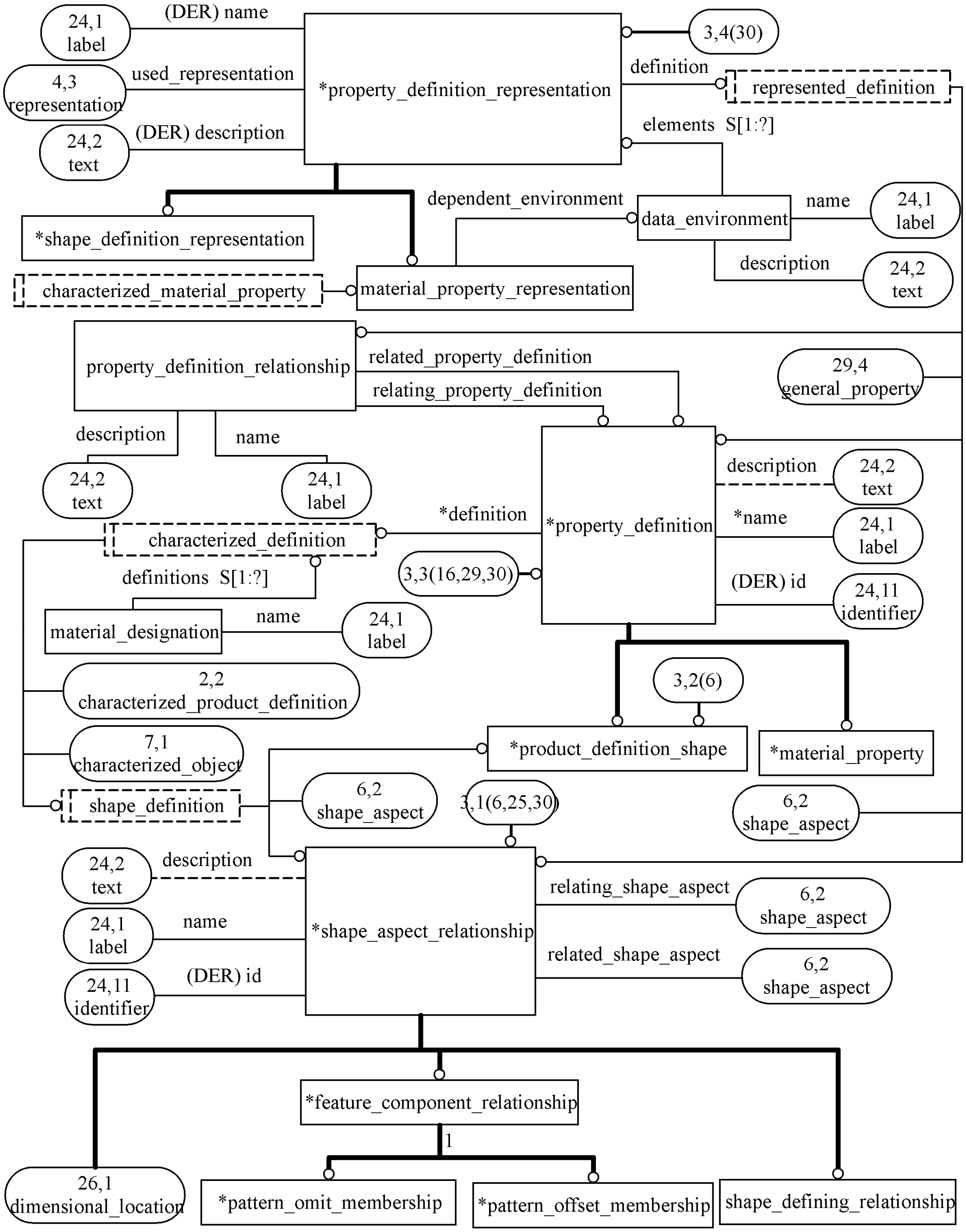


Figure H.3 — property definition — AIM EXPRESS-G diagram 3 of 30

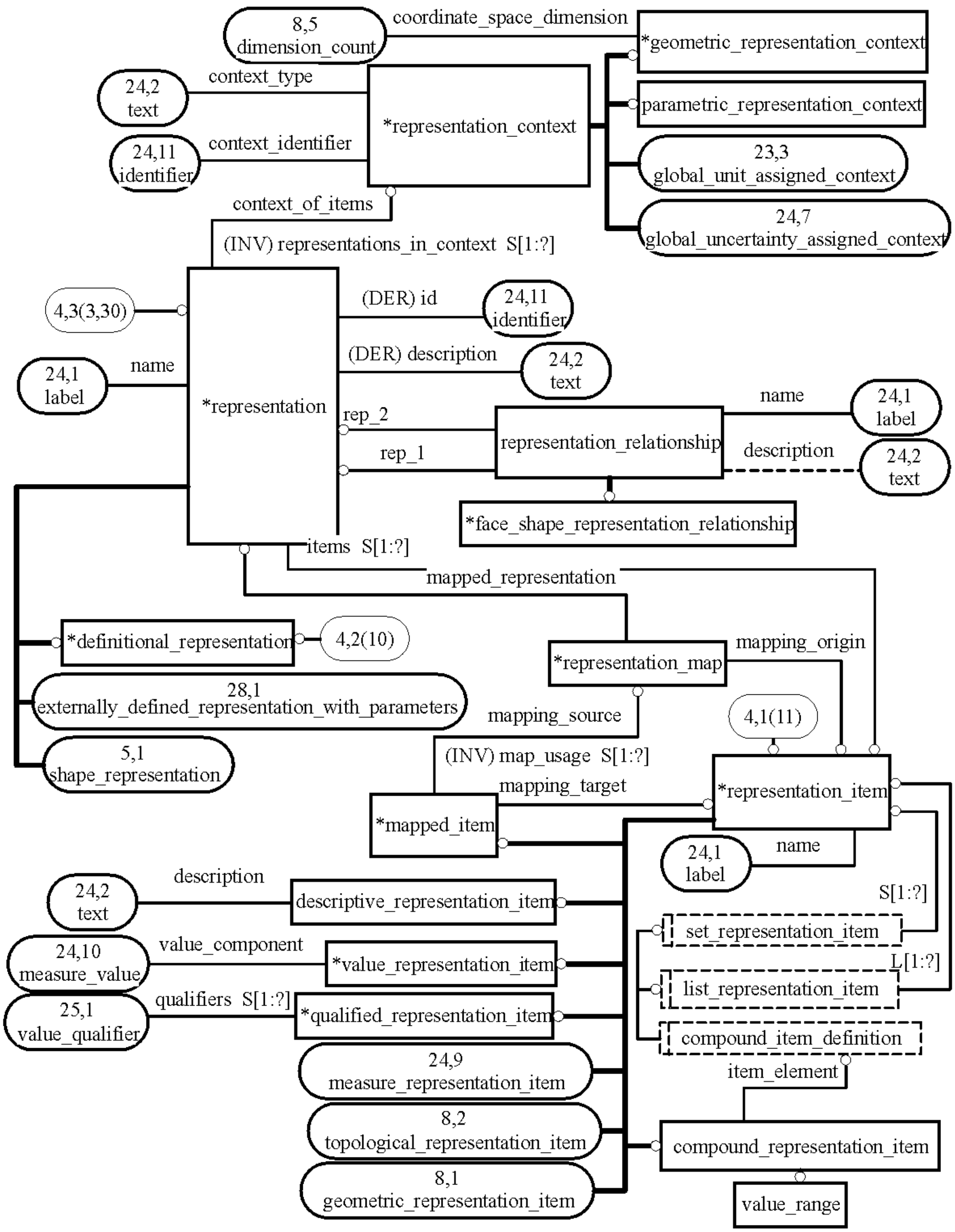


Figure H.4 — representation — AIM EXPRESS-G diagram 4 of 30

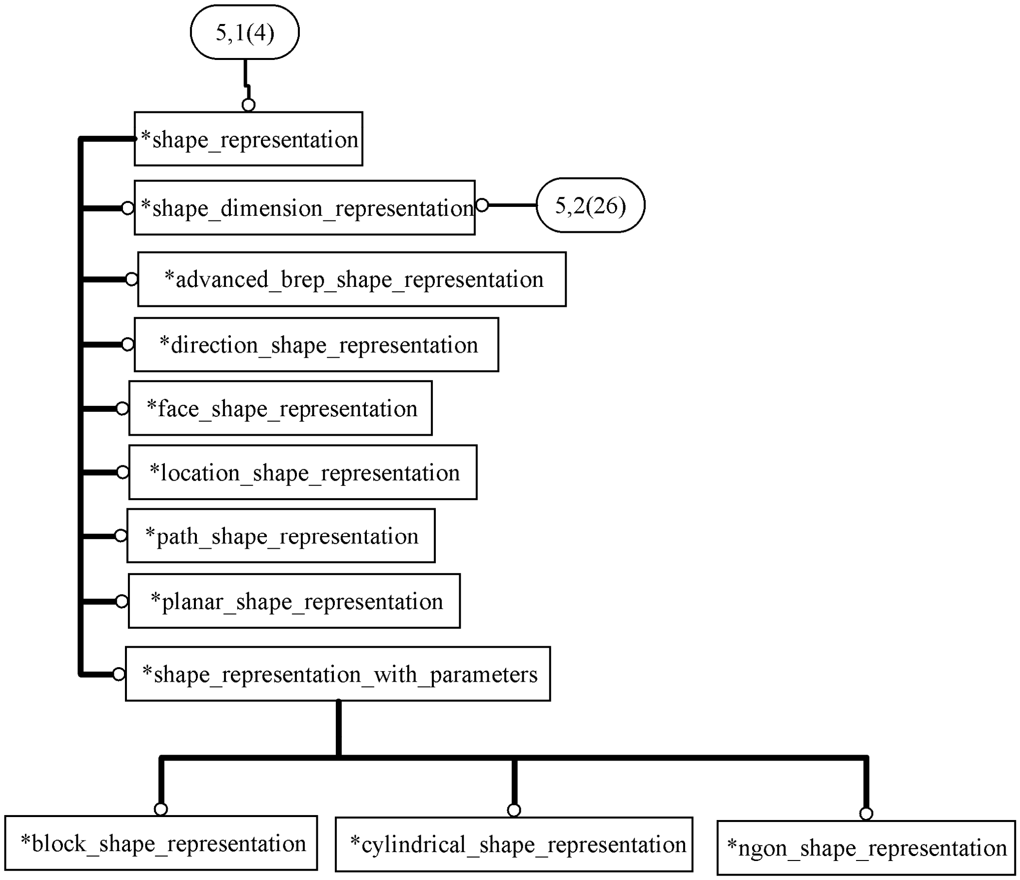


Figure H.5 — shape_representation — AIM EXPRESS-G diagram 5 of 30

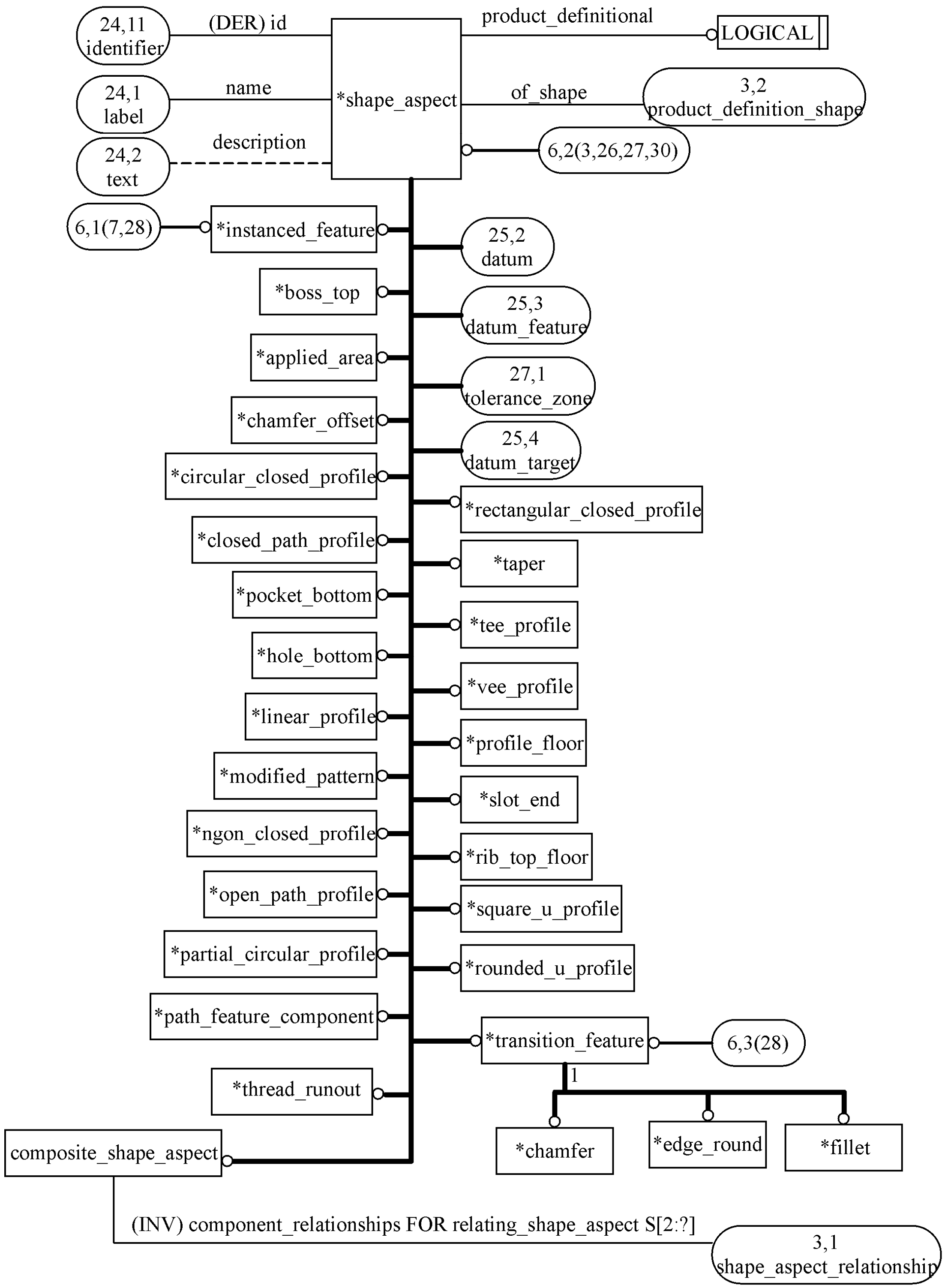


Figure H.6 — shape_aspect — AIM EXPRESS-G diagram 6 of 30

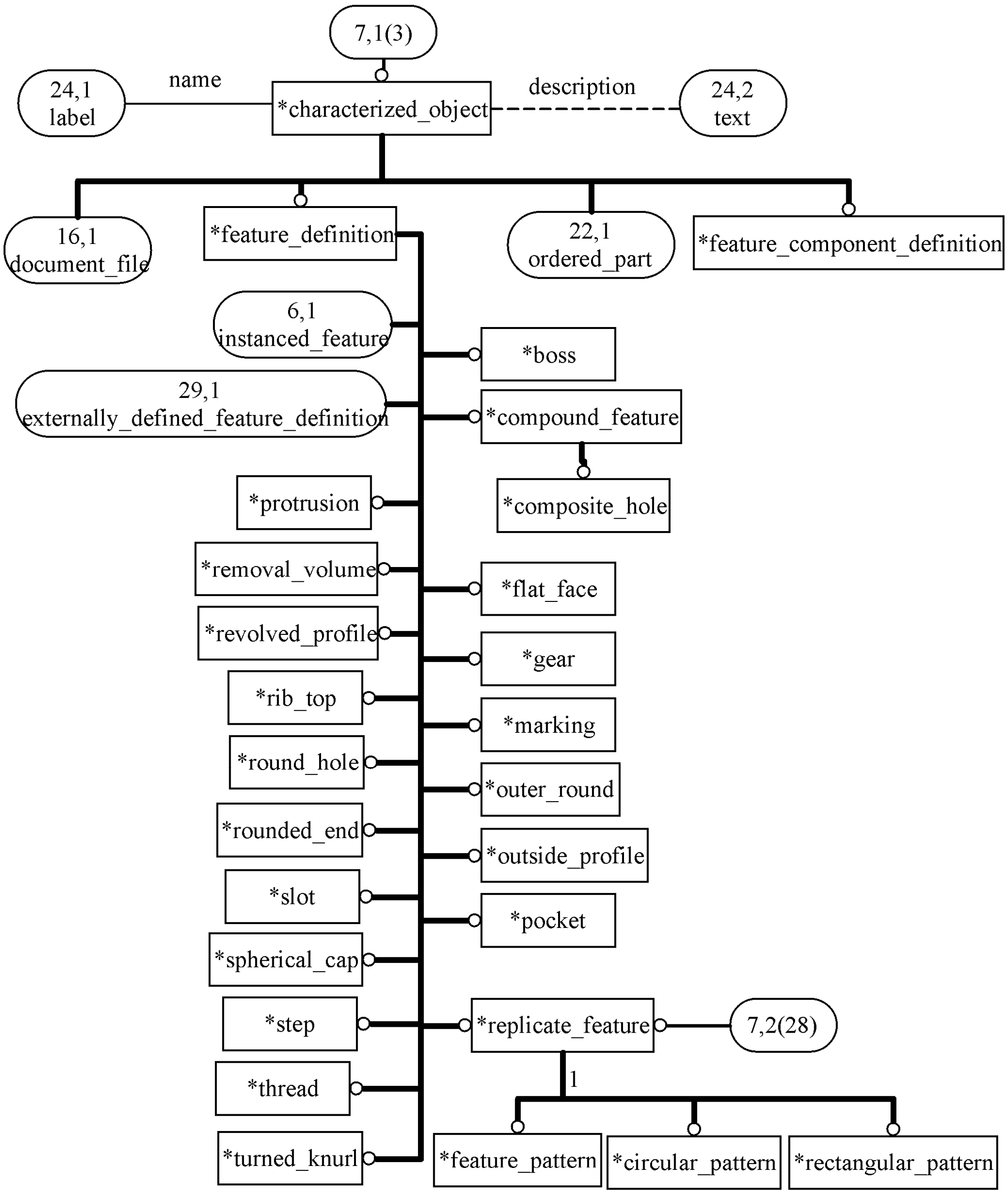


Figure H.7 — characterized_object — AIM EXPRESS-G diagram 7 of 30

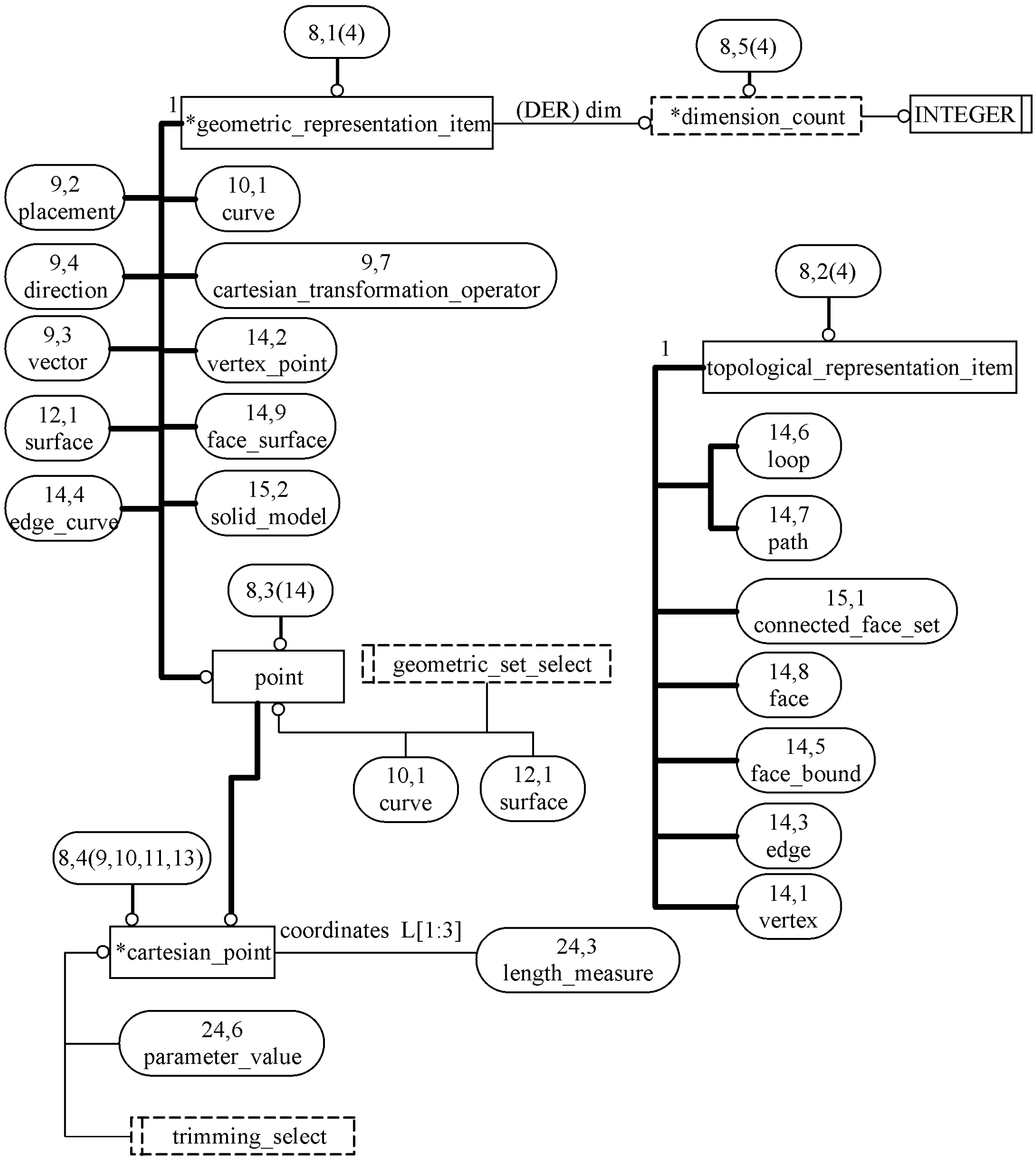


Figure H.8 — geometry topology — AIM EXPRESS-G diagram 8 of 30

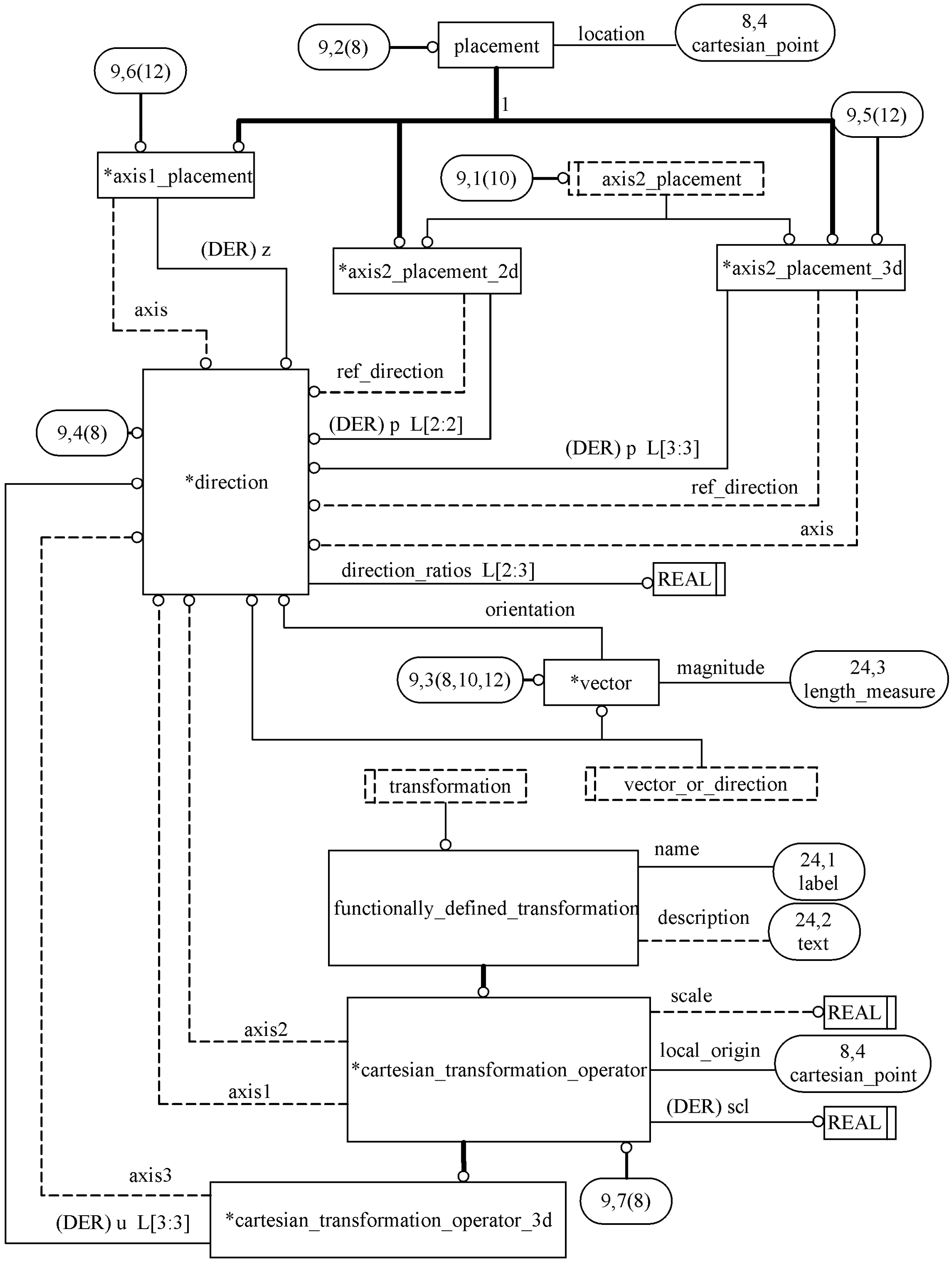


Figure H.9 — geometric orientation — AIM EXPRESS-G diagram 9 of 30

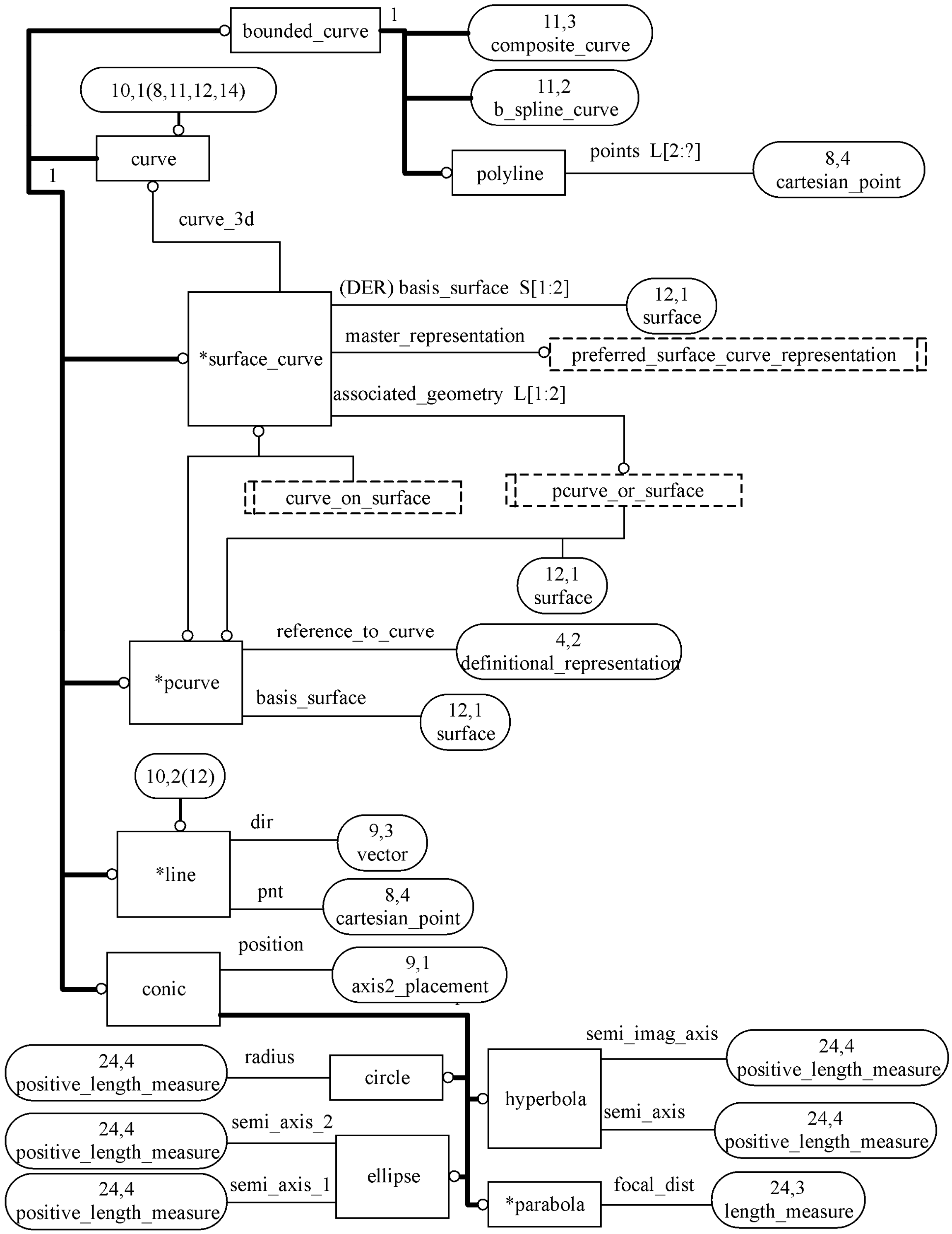


Figure H.10 — curve — AIM EXPRESS-G diagram 10 of 30

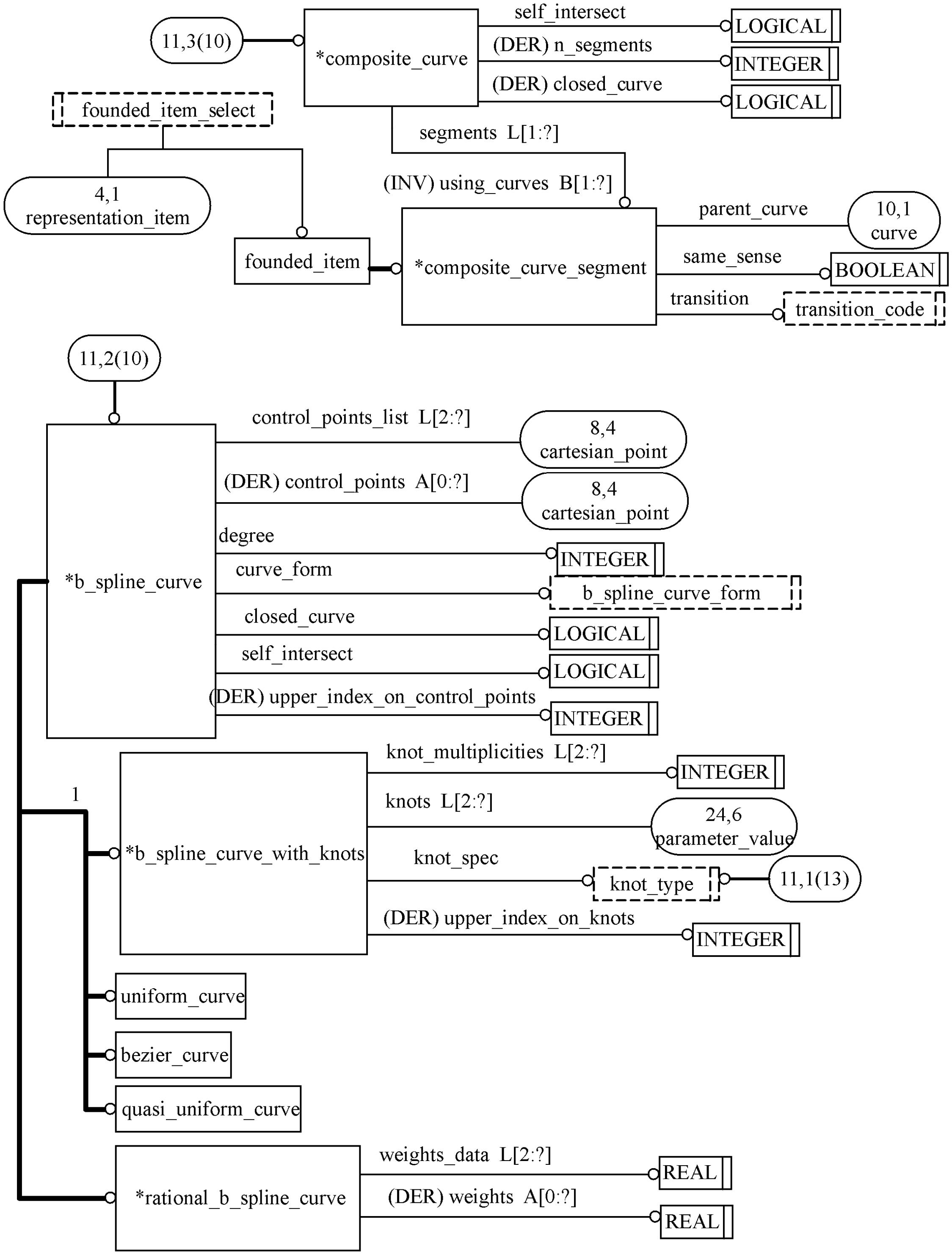


Figure H.11 — bounded_curve — AIM EXPRESS-G diagram 11 of 30

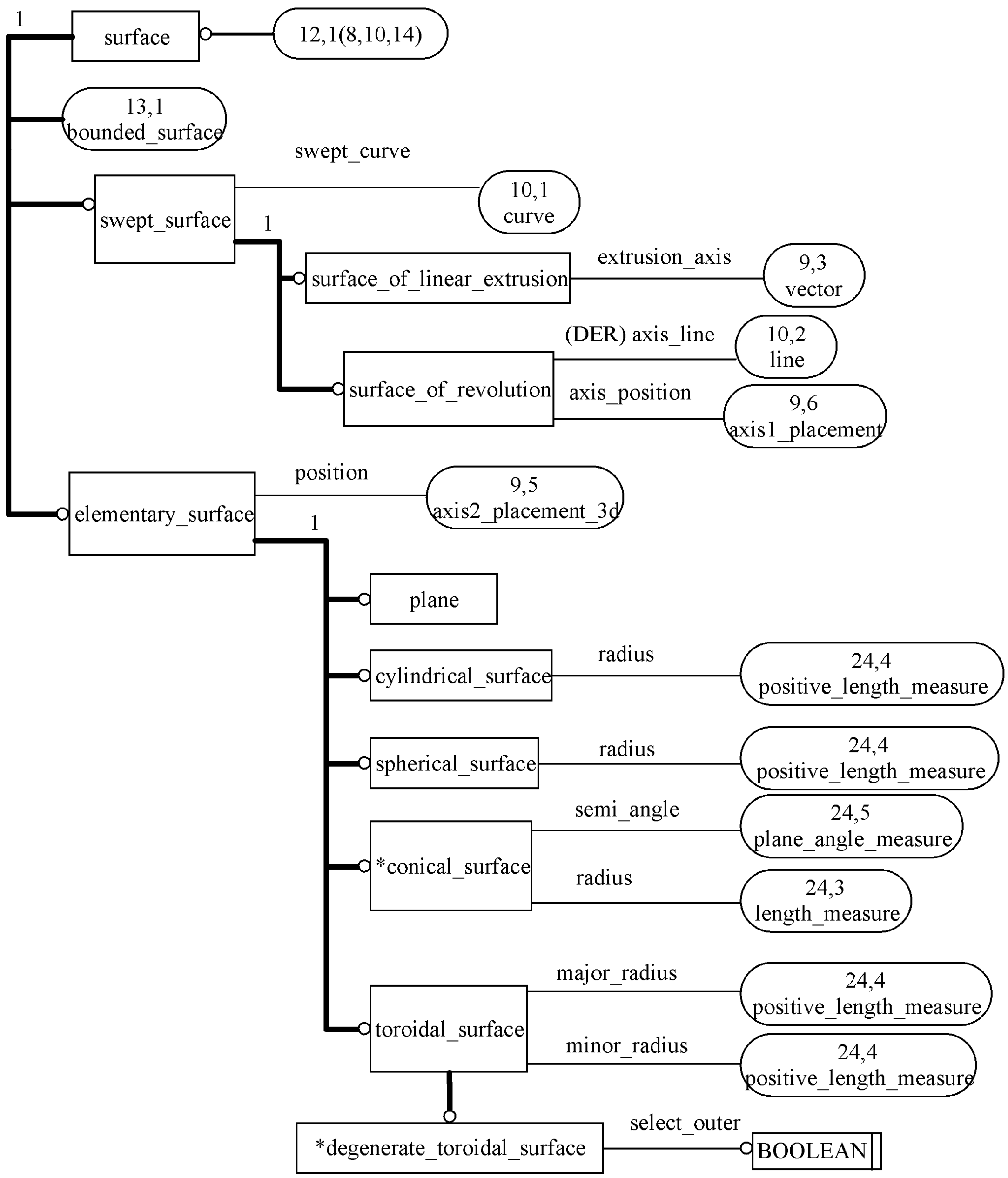


Figure H.12 — surface — AIM EXPRESS-G diagram 12 of 30

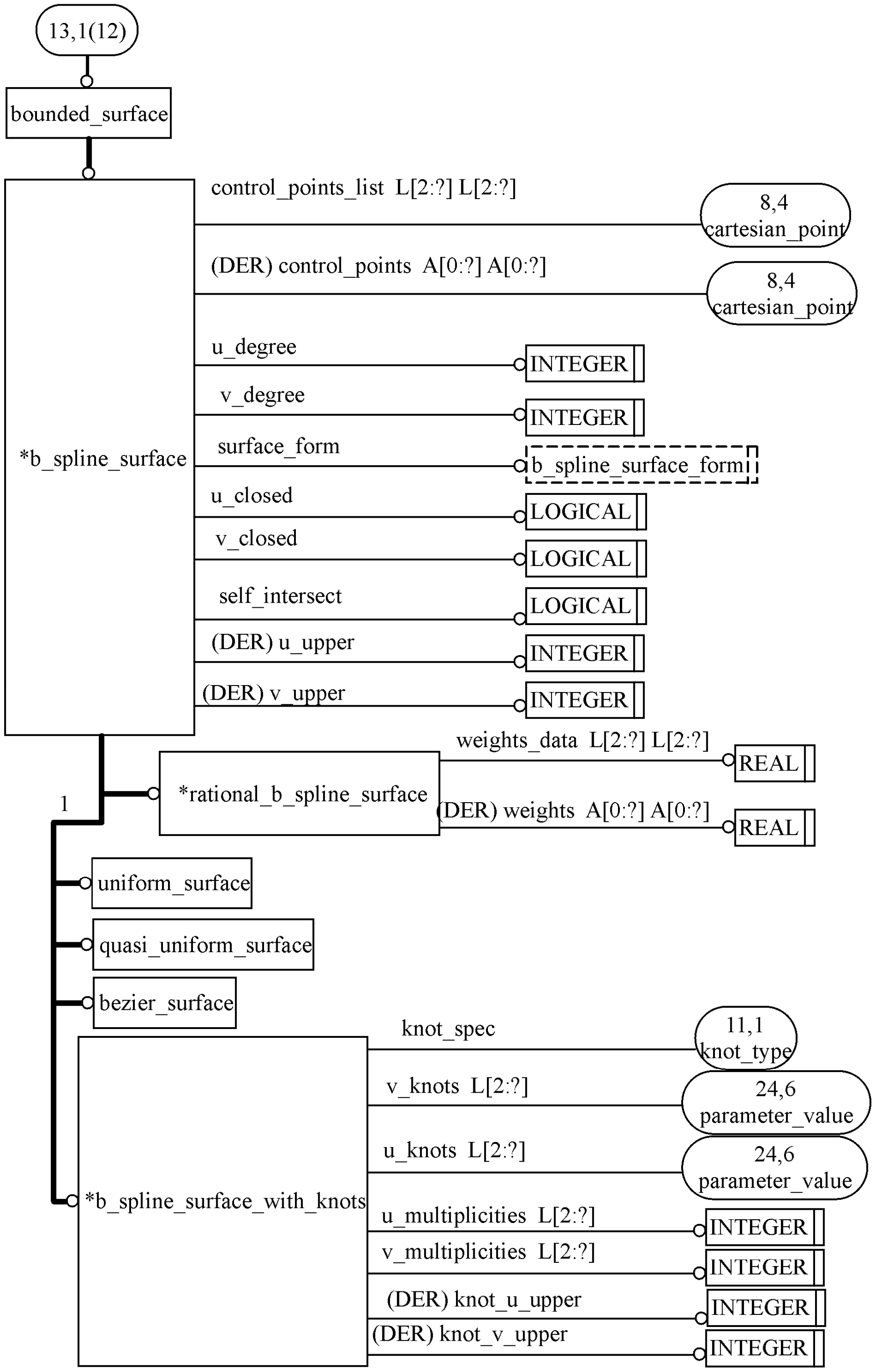


Figure H.13 — bounded_surface — AIM EXPRESS-G diagram 13 of 30

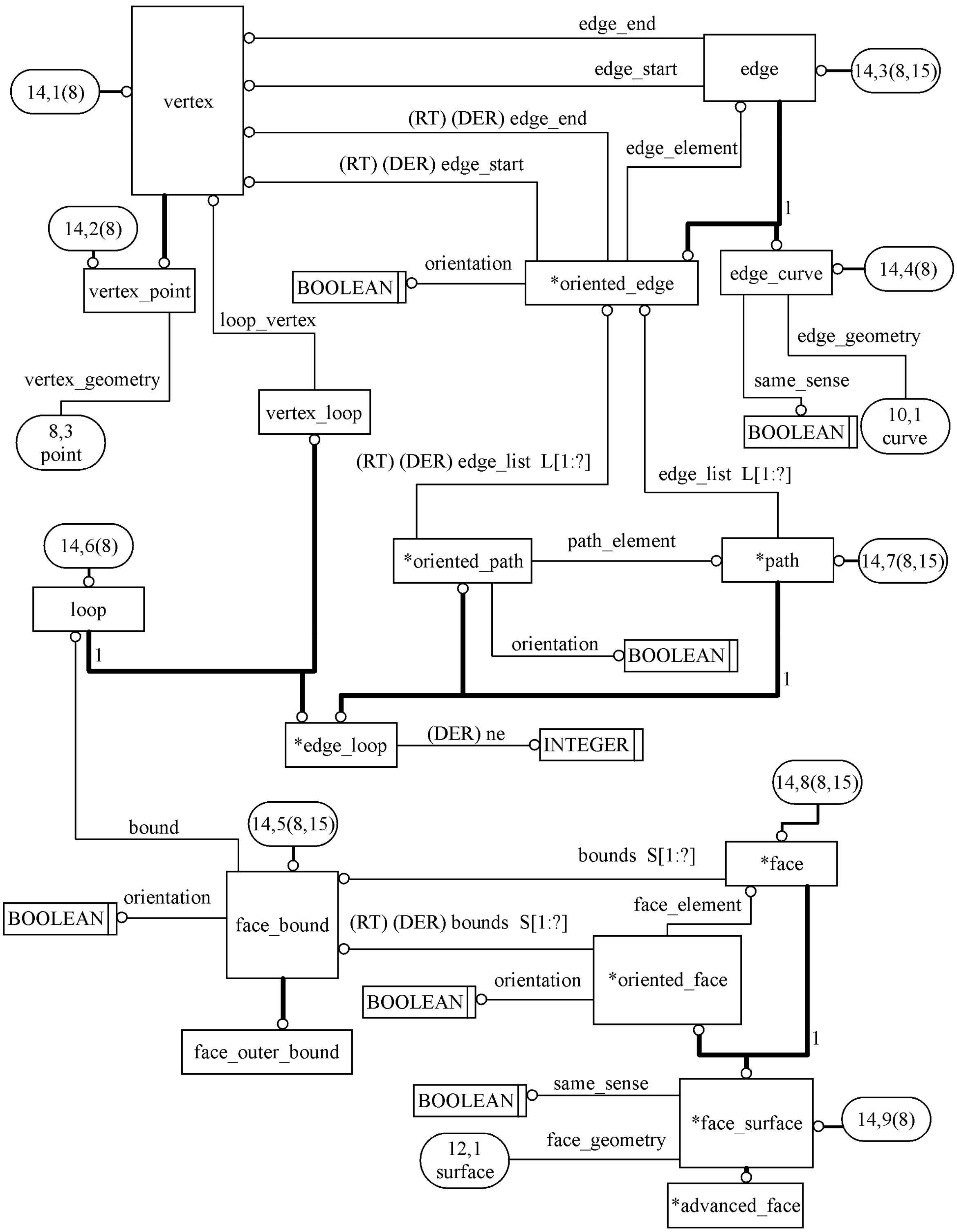


Figure H.14 — topology — AIM EXPRESS-G diagram 14 of 30

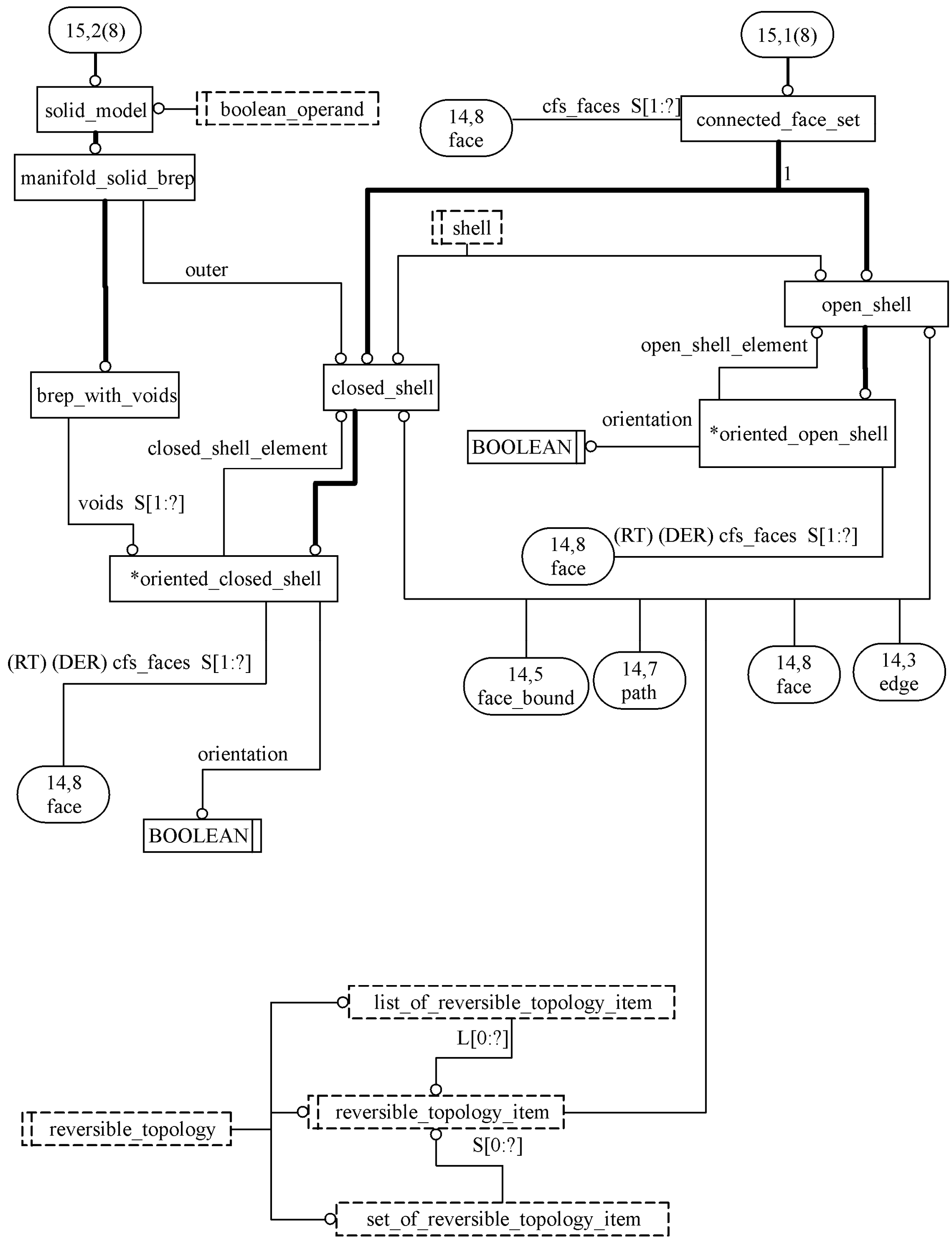


Figure H.15 — shell — AIM EXPRESS-G diagram 15 of 30

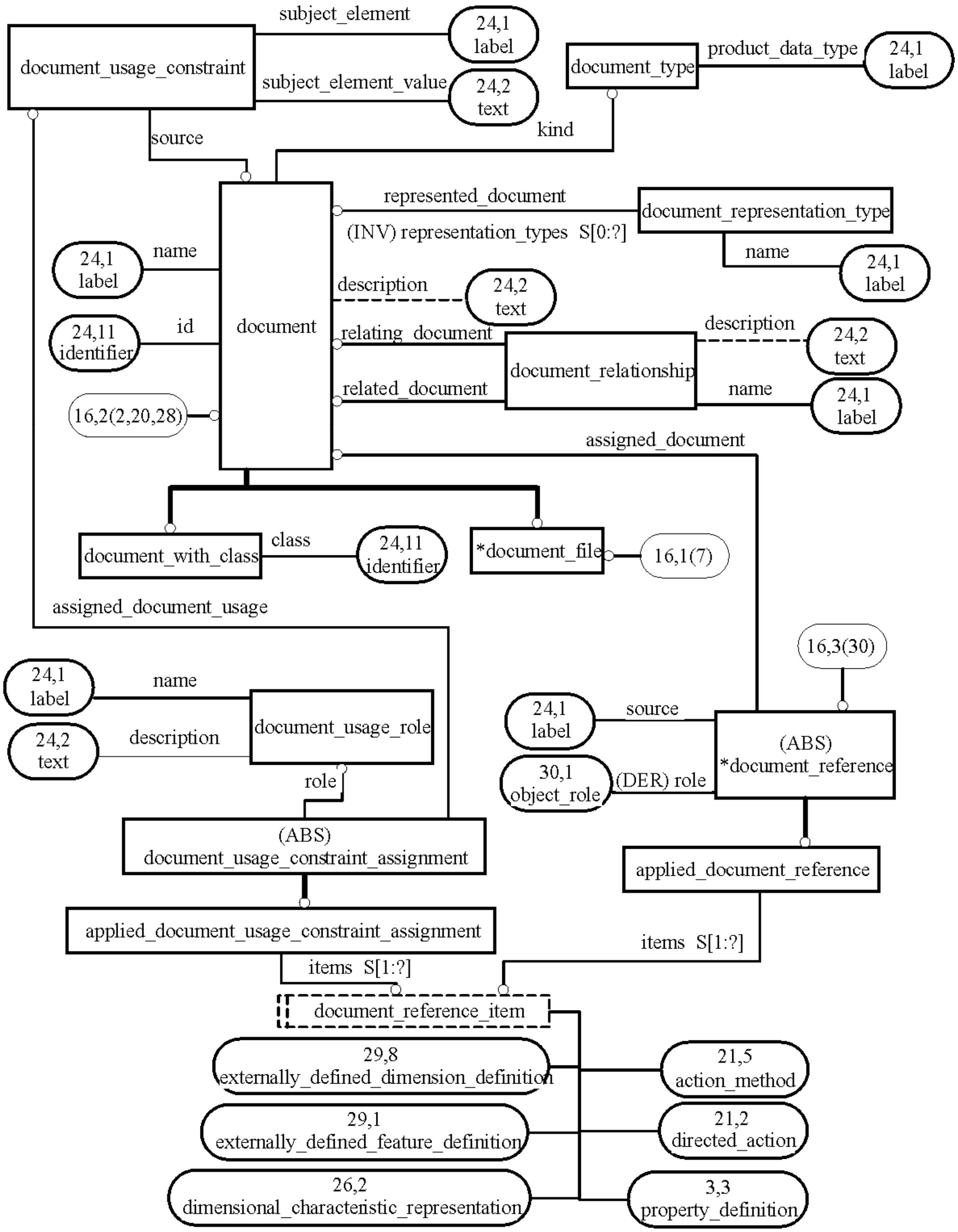


Figure H.16 — document — AIM EXPRESS-G diagram 16 of 30

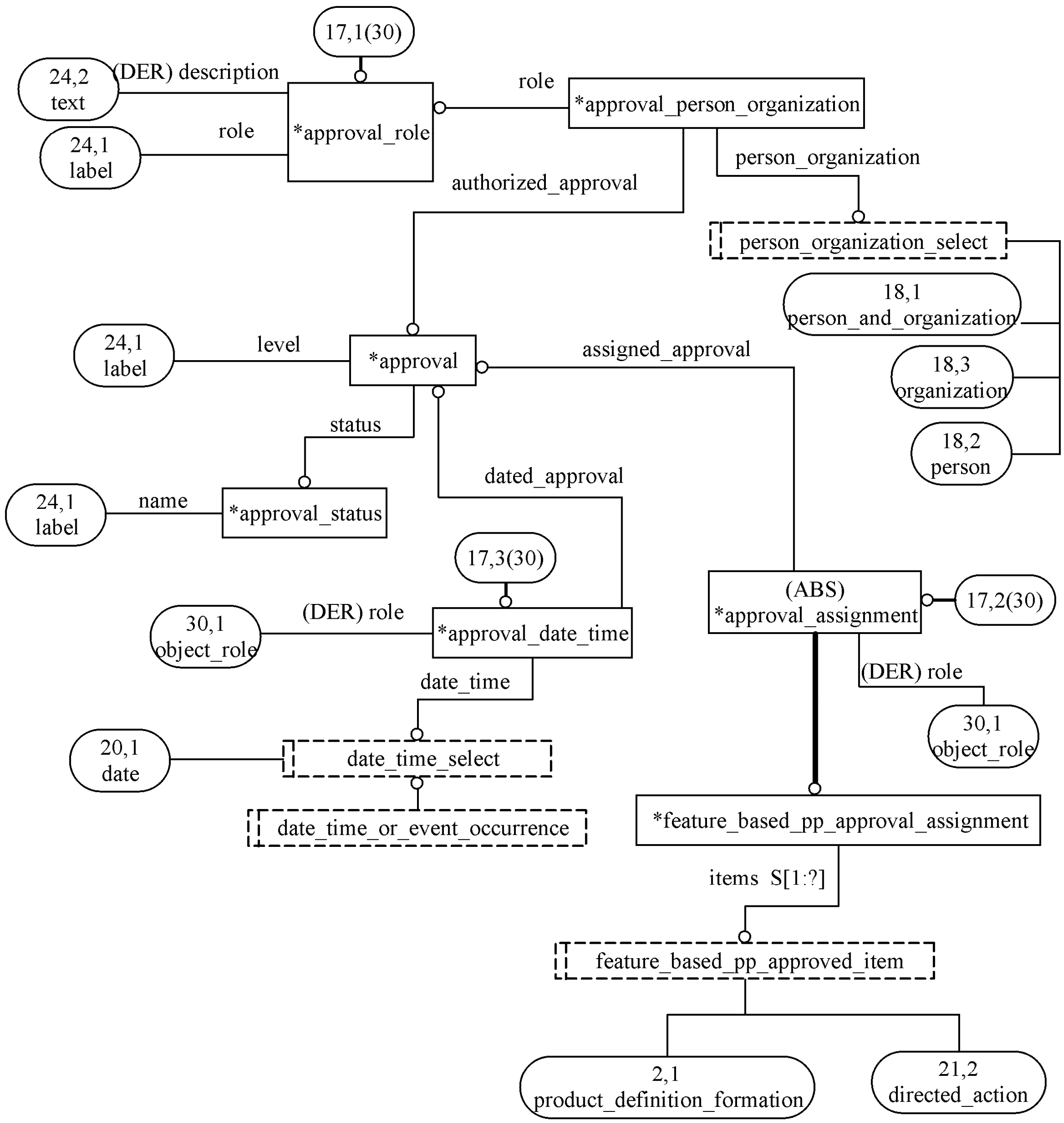


Figure H.17 — approval — AIM EXPRESS-G diagram 17 of 30

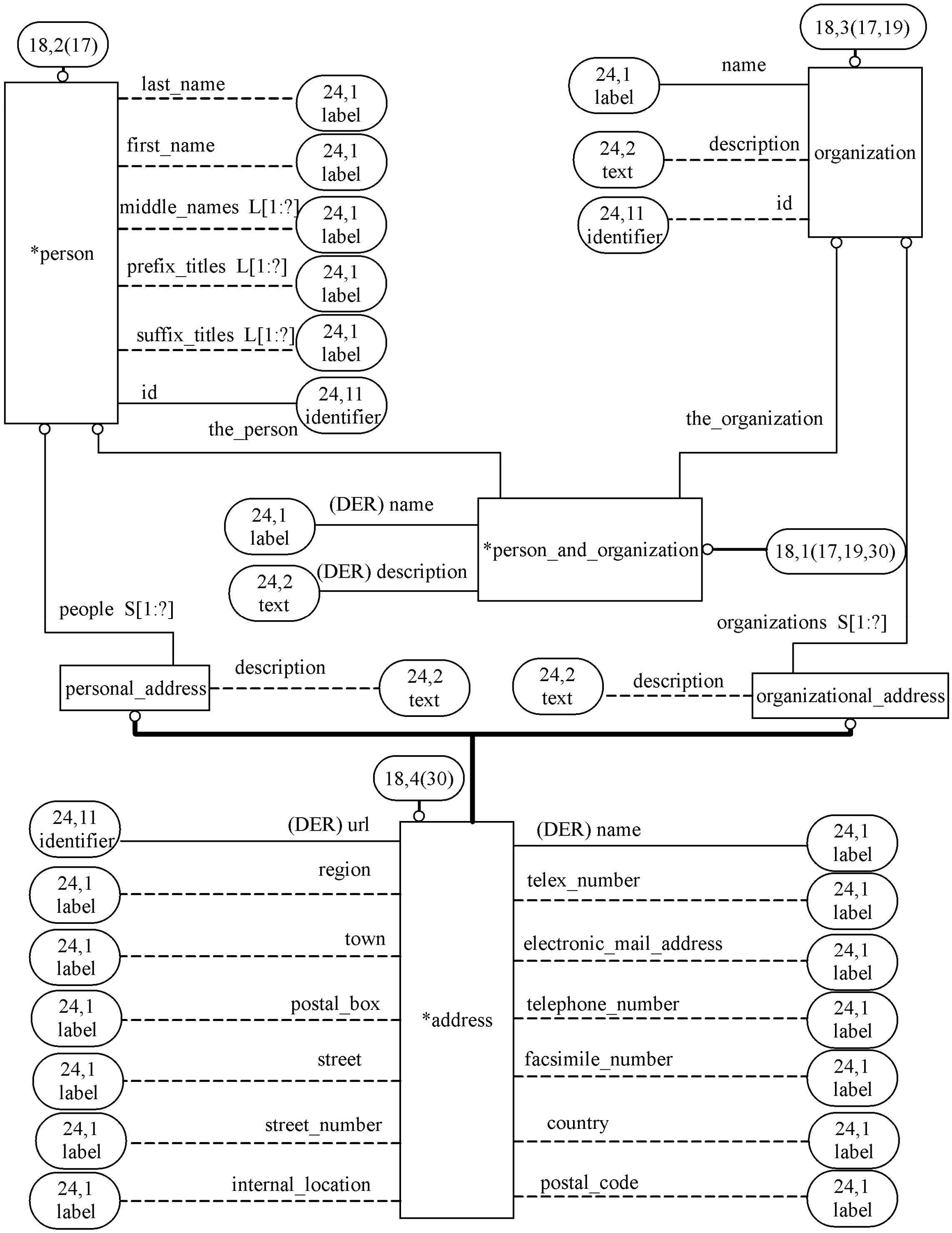


Figure H.18 — person organization — AIM EXPRESS-G diagram 18 of 30

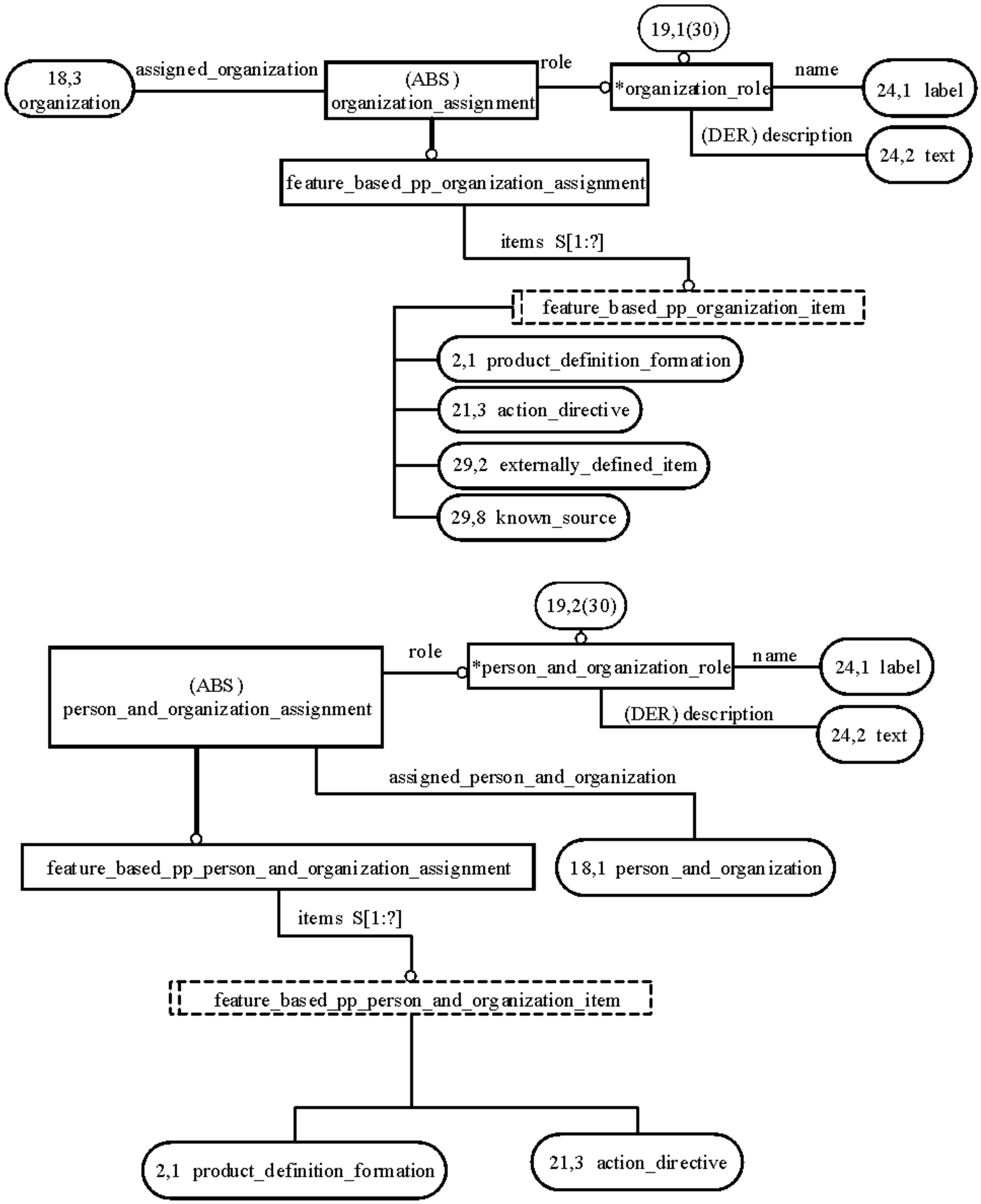


Figure H.19 — person organization assignment — AIM EXPRESS-G diagram 19 of 30

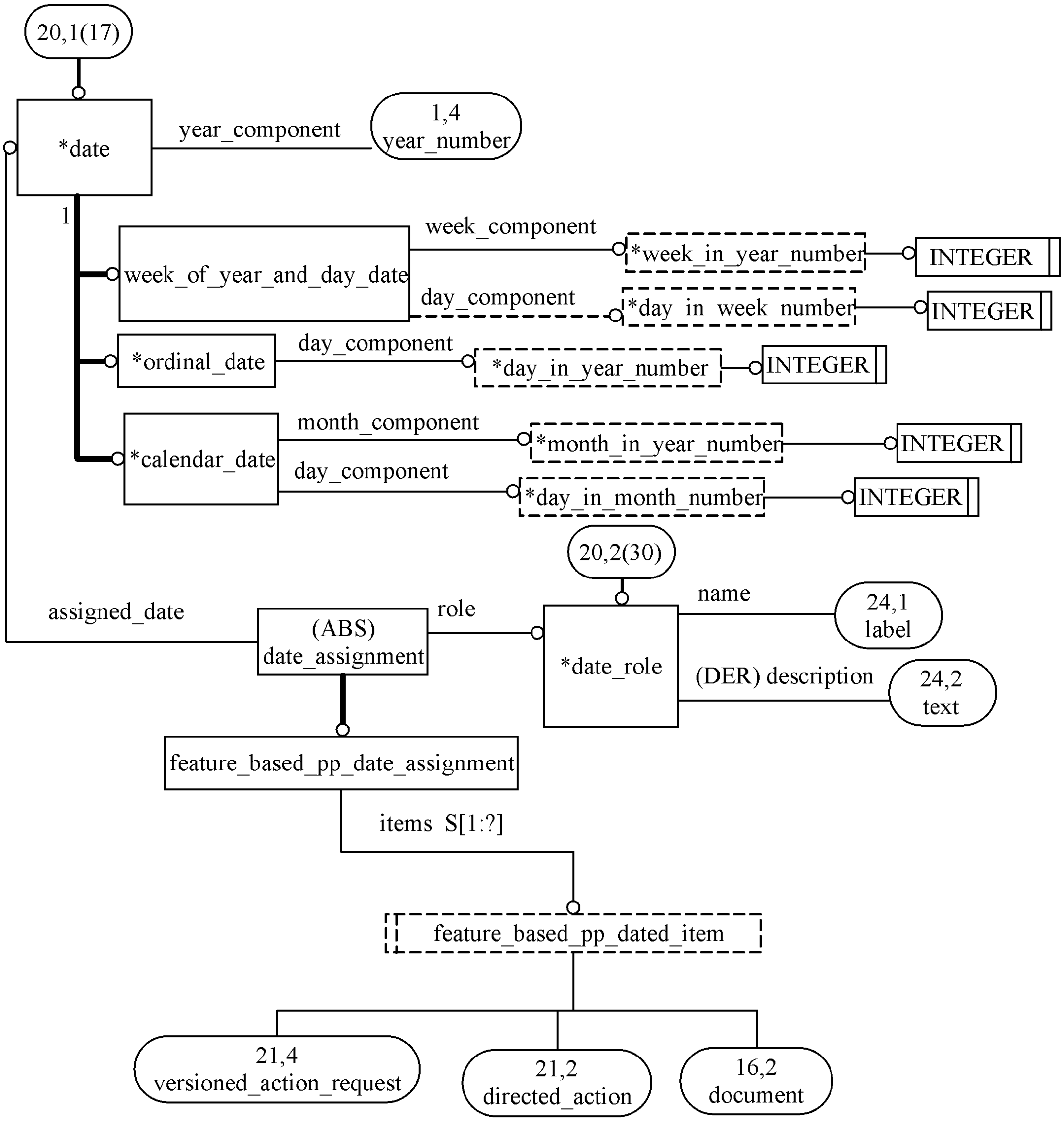


Figure H.20 — date — AIM EXPRESS-G diagram 20 of 30

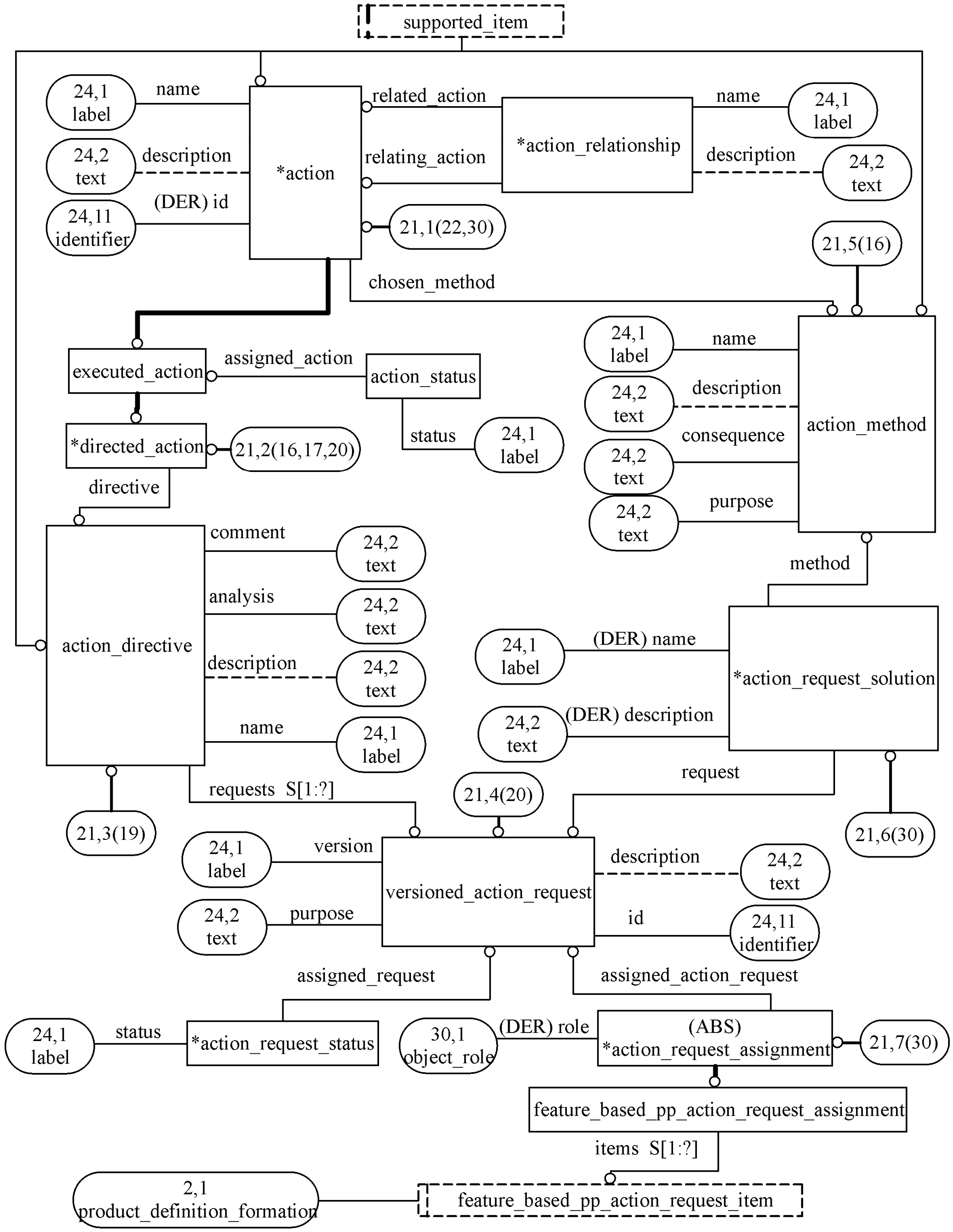


Figure H.21 — action — AIM EXPRESS-G diagram 21 of 30

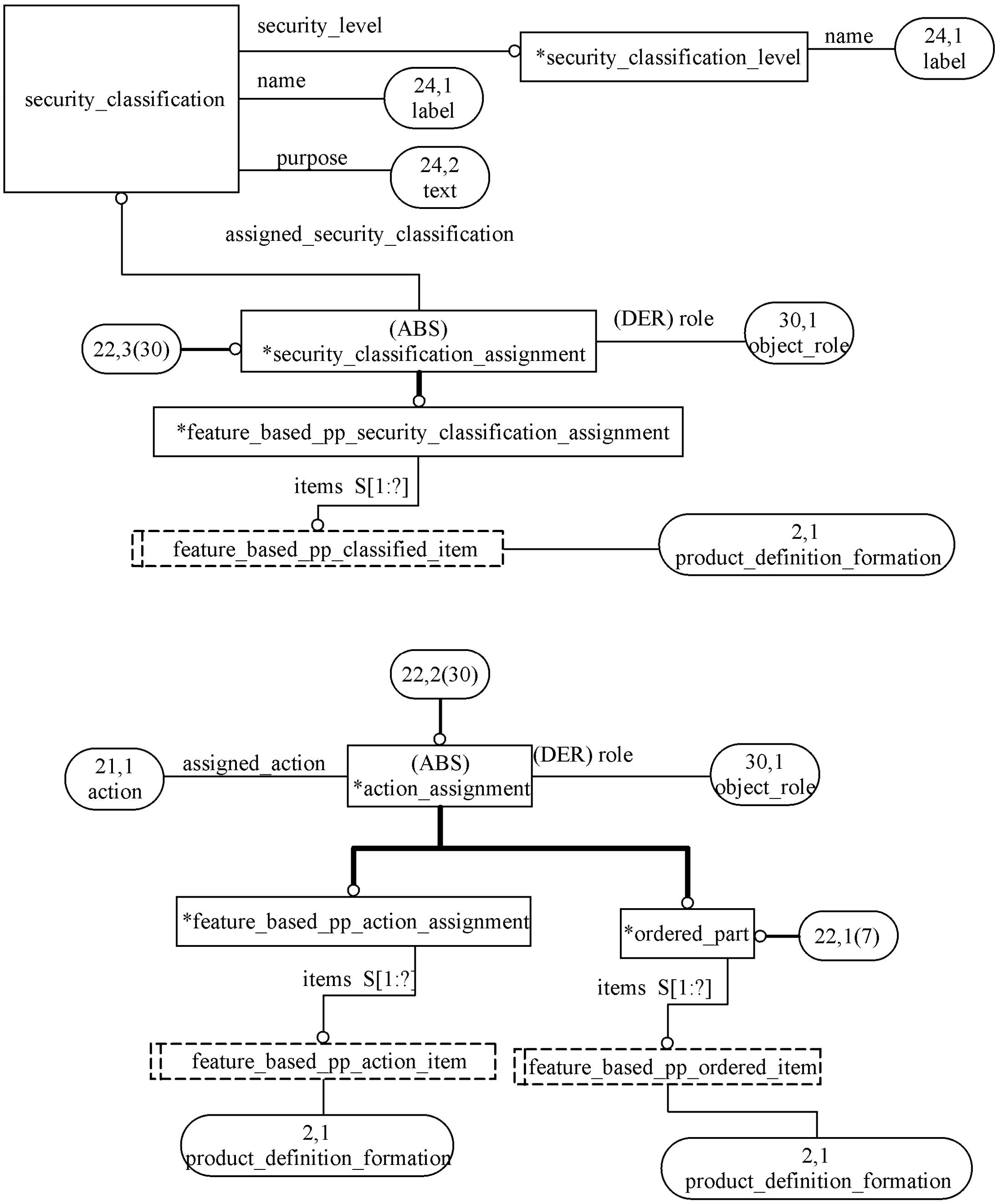


Figure H.22 — security classification — AIM EXPRESS-G diagram 22 of 30

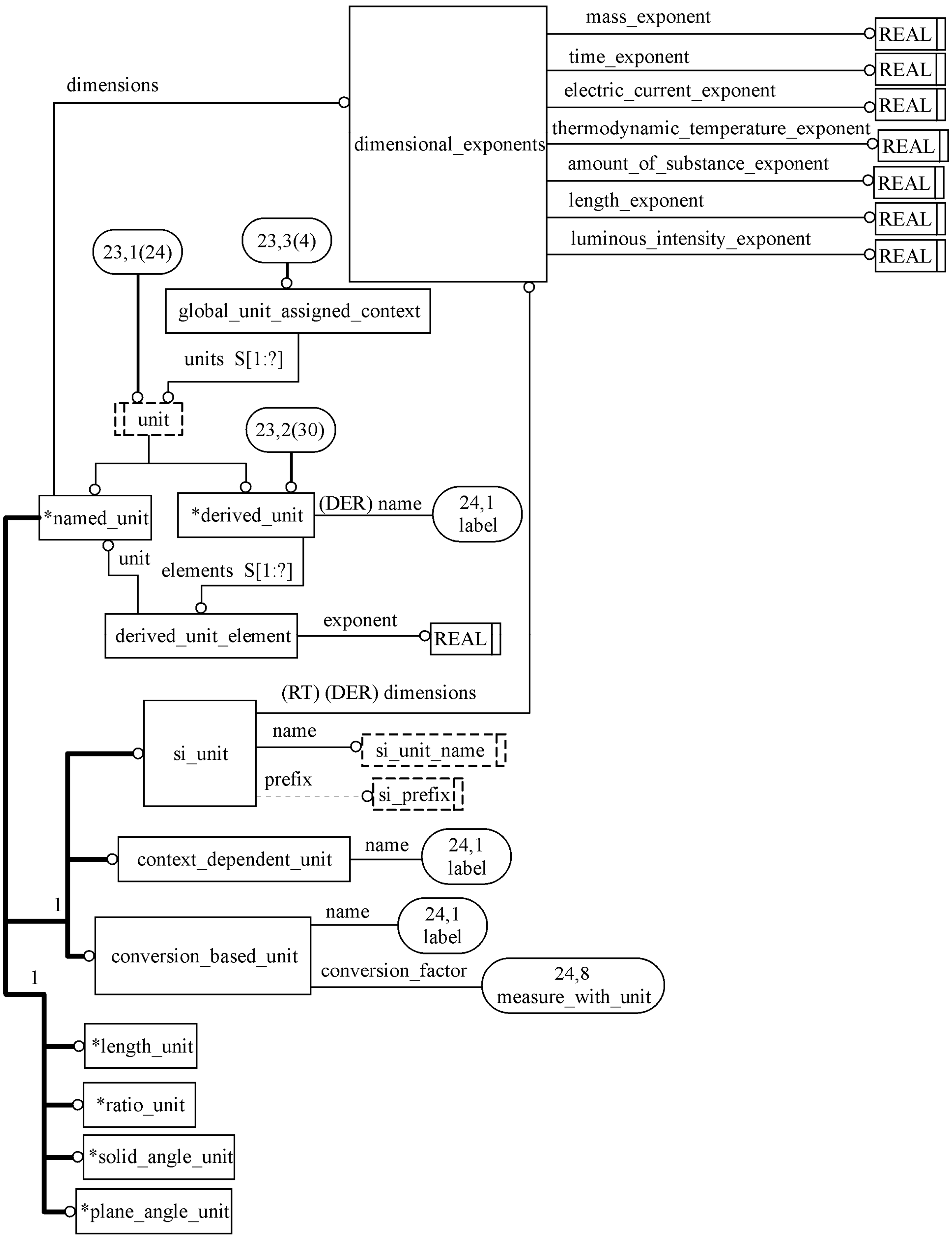


Figure H.23 — units — AIM EXPRESS-G diagram 23 of 30

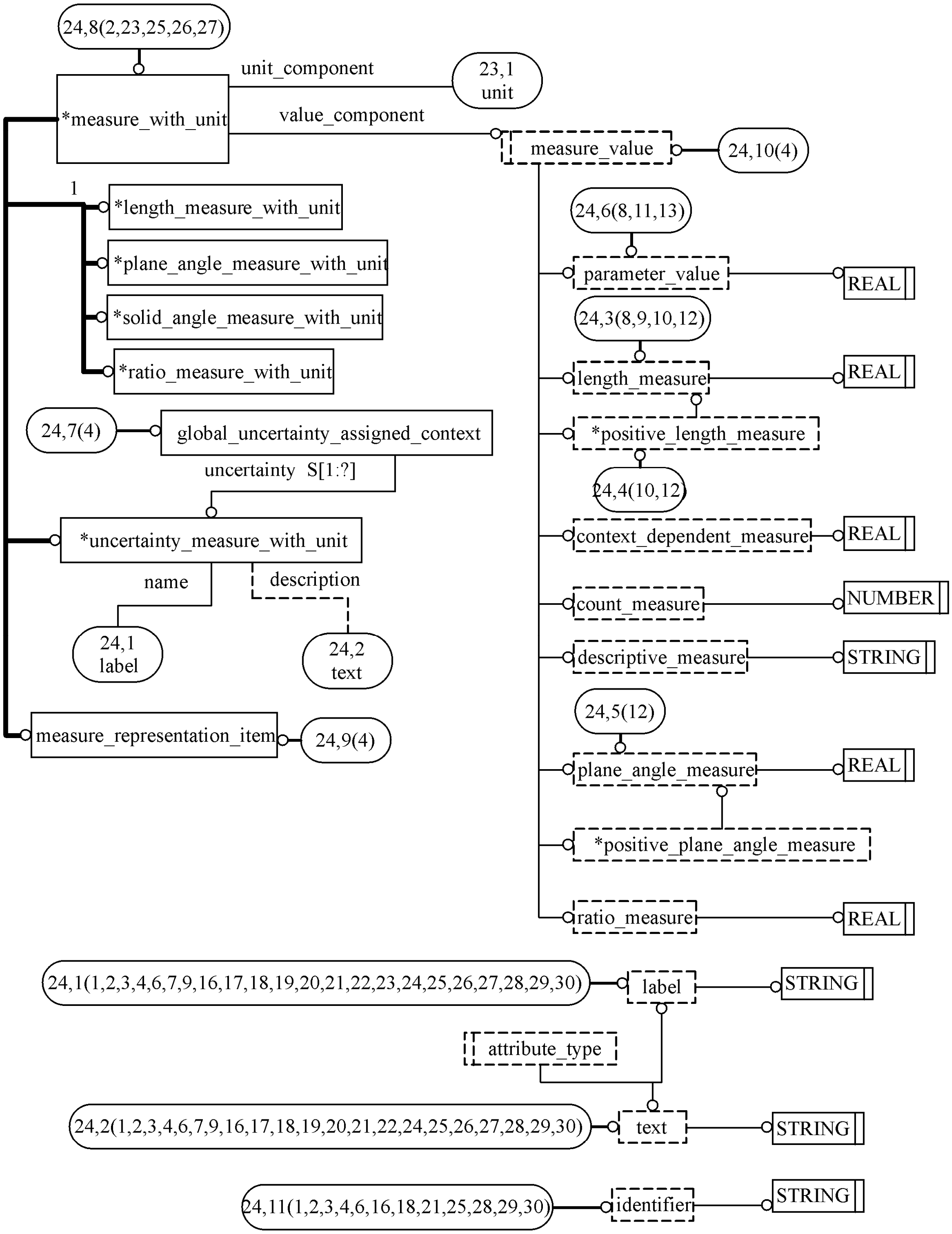


Figure H.24 — measure with unit — AIM EXPRESS-G diagram 24 of 30

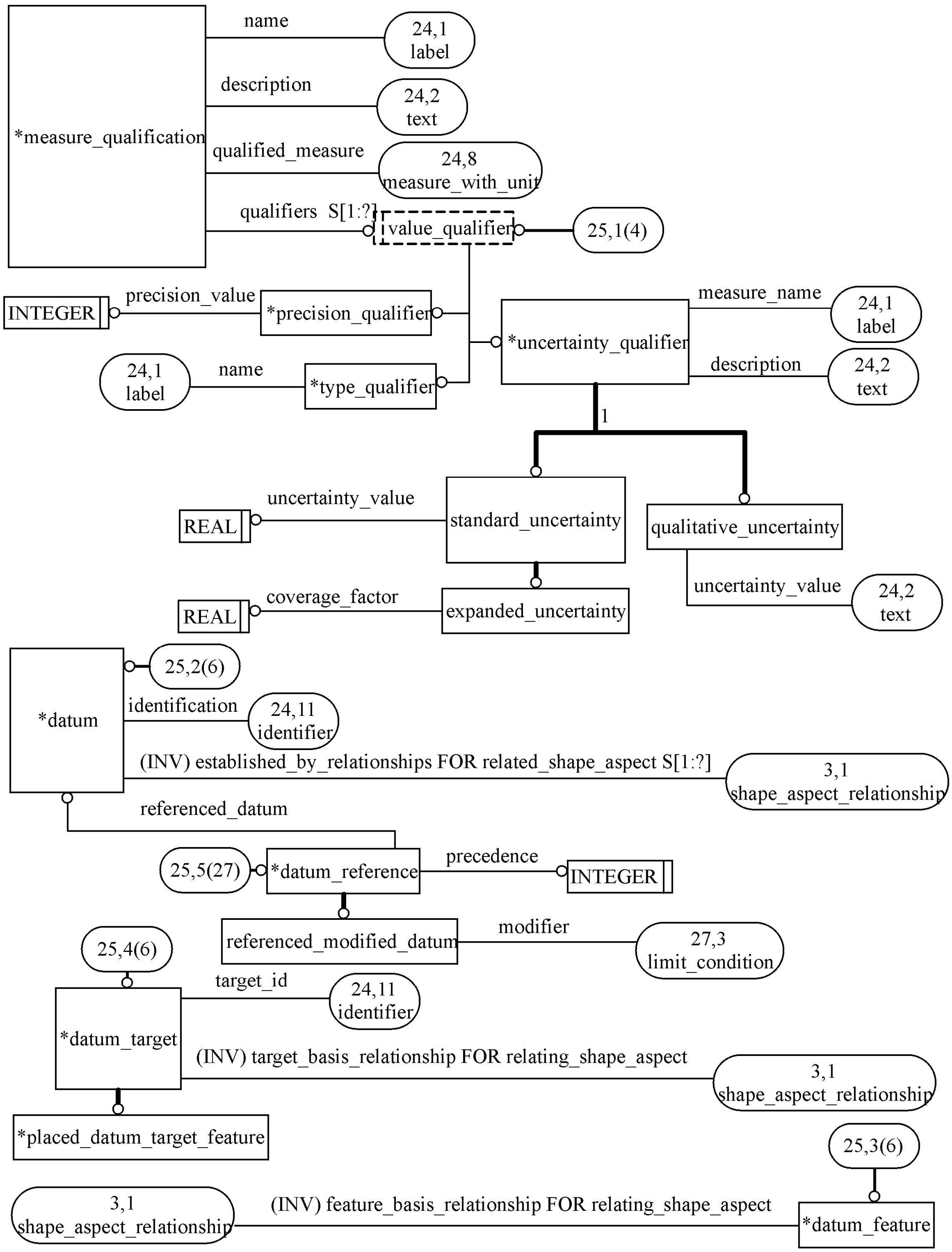


Figure H.25 — measure qualification and datums — AIM EXPRESS-G diagram 25 of 30

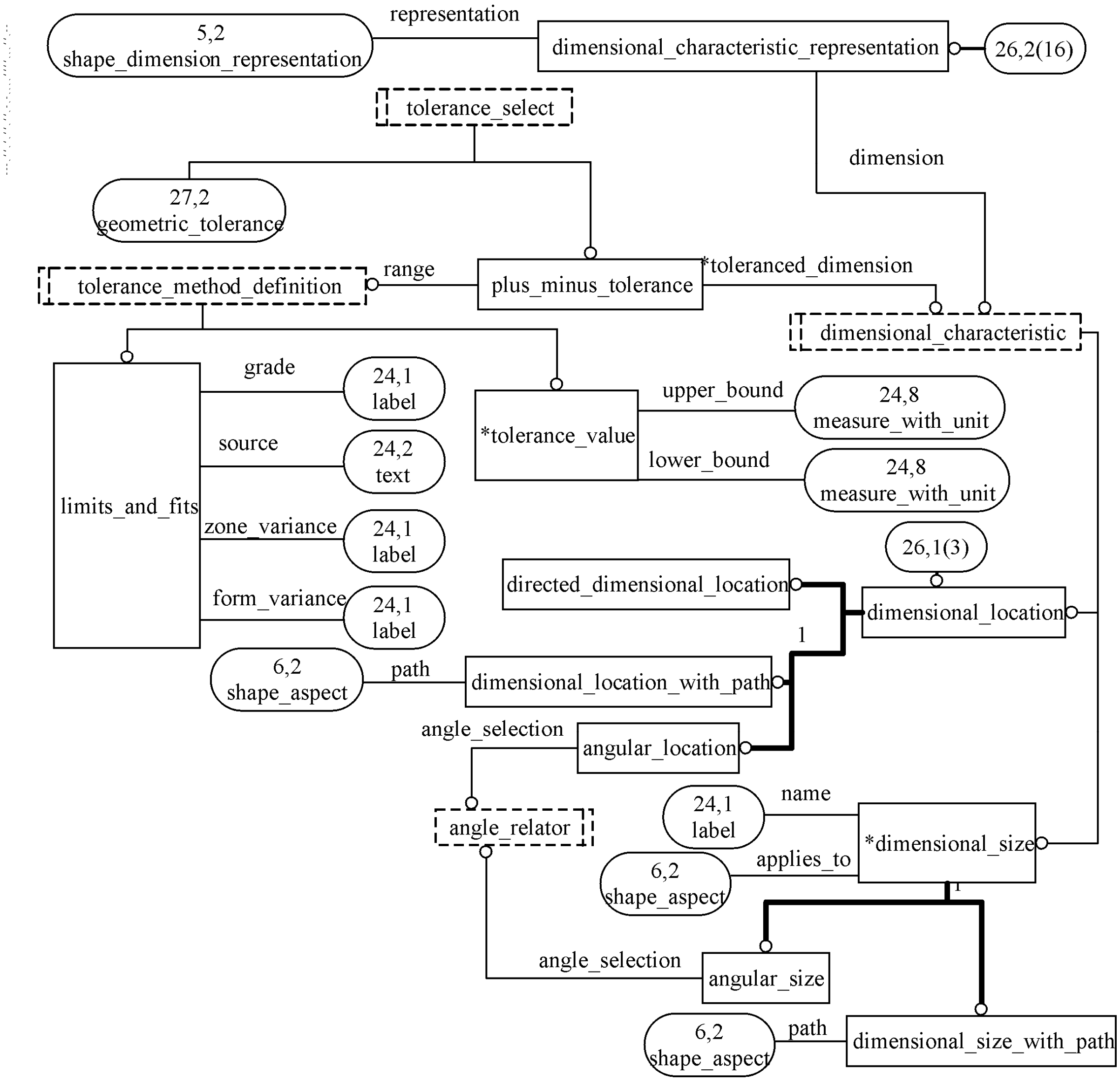


Figure H.26 — tolerances — AIM EXPRESS-G diagram 26 of 30

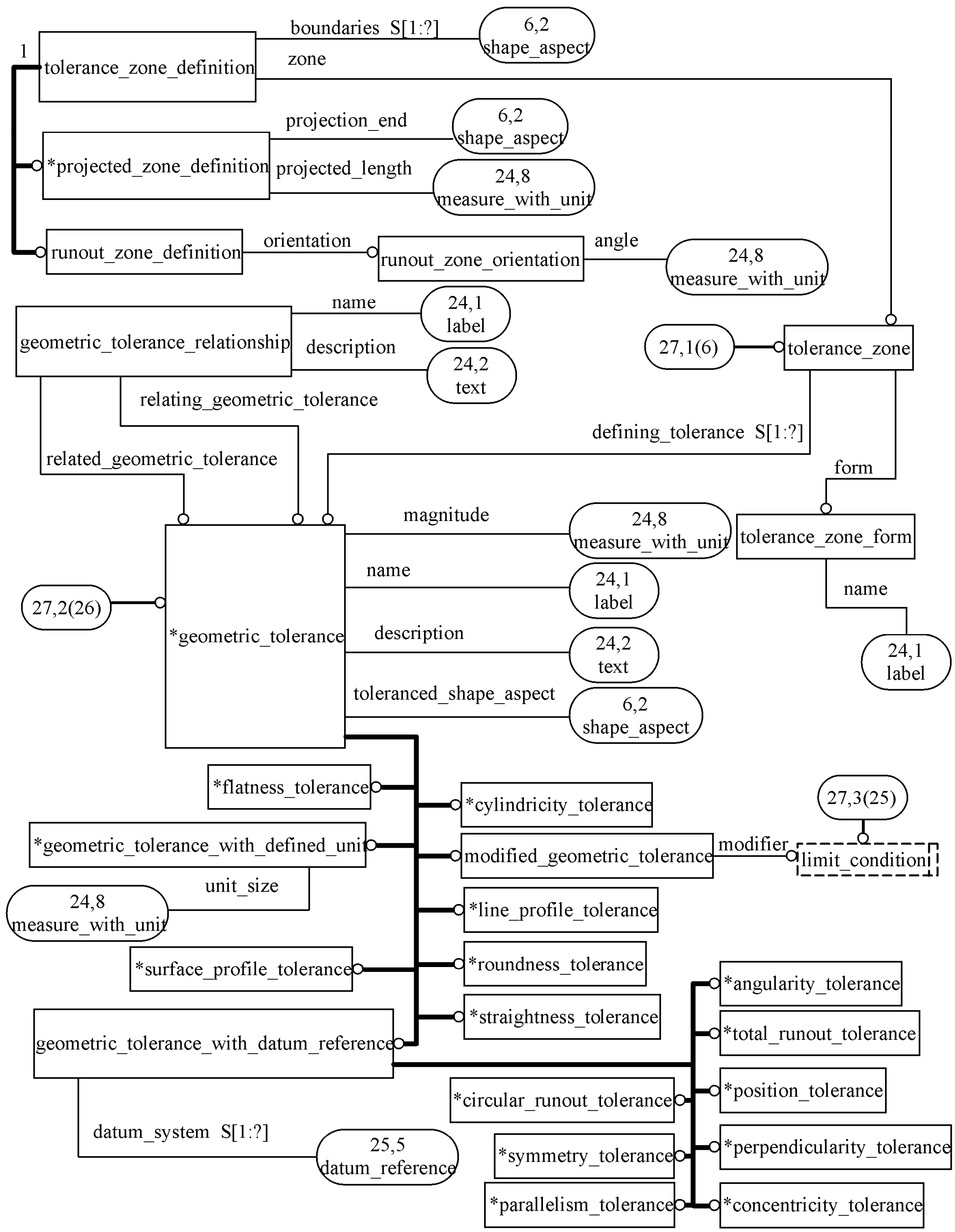


Figure H.27 — geometric tolerance — AIM EXPRESS-G diagram 27 of 30

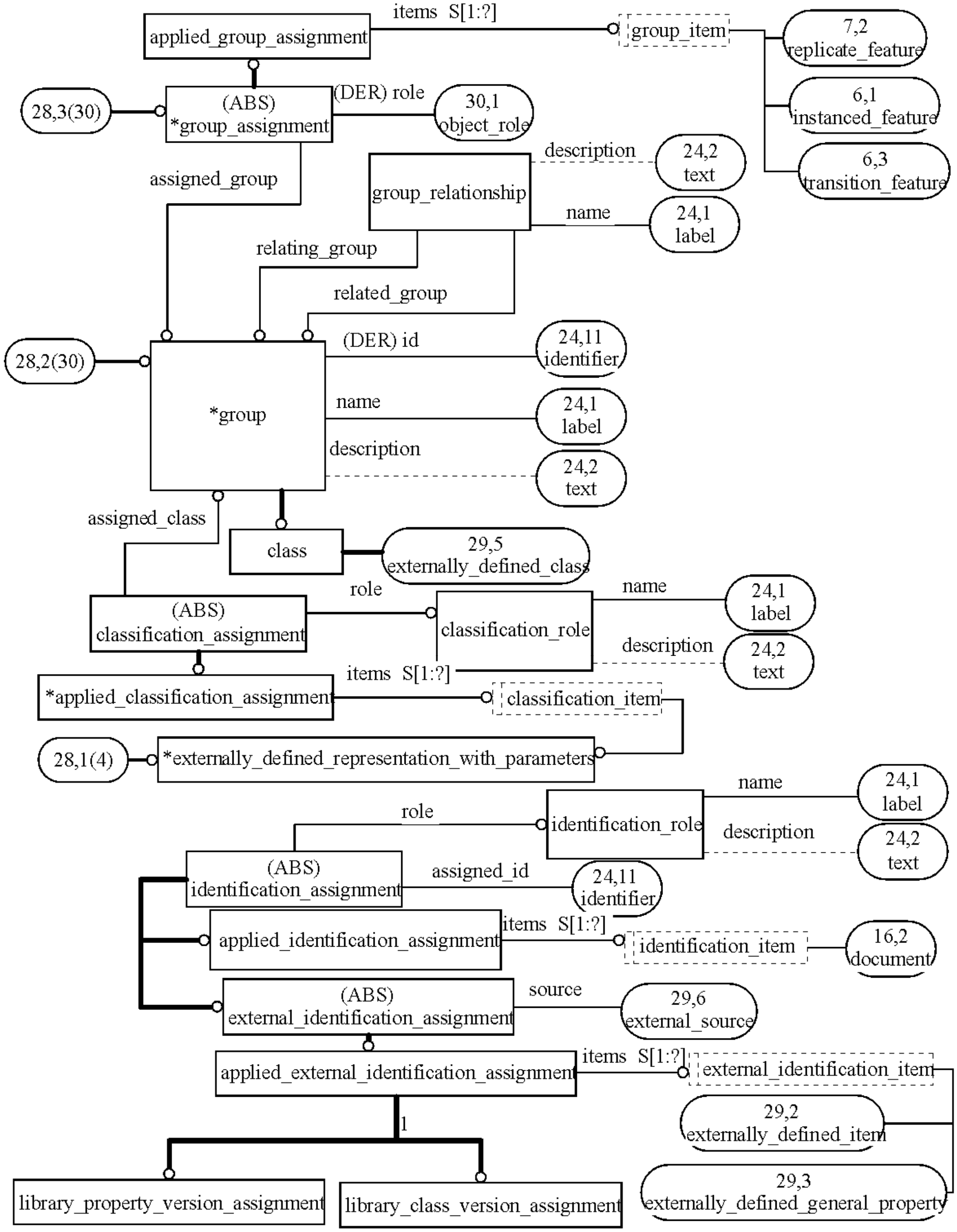


Figure H.28 — group and identification assignment — AIM EXPRESS-G diagram 28 of 30

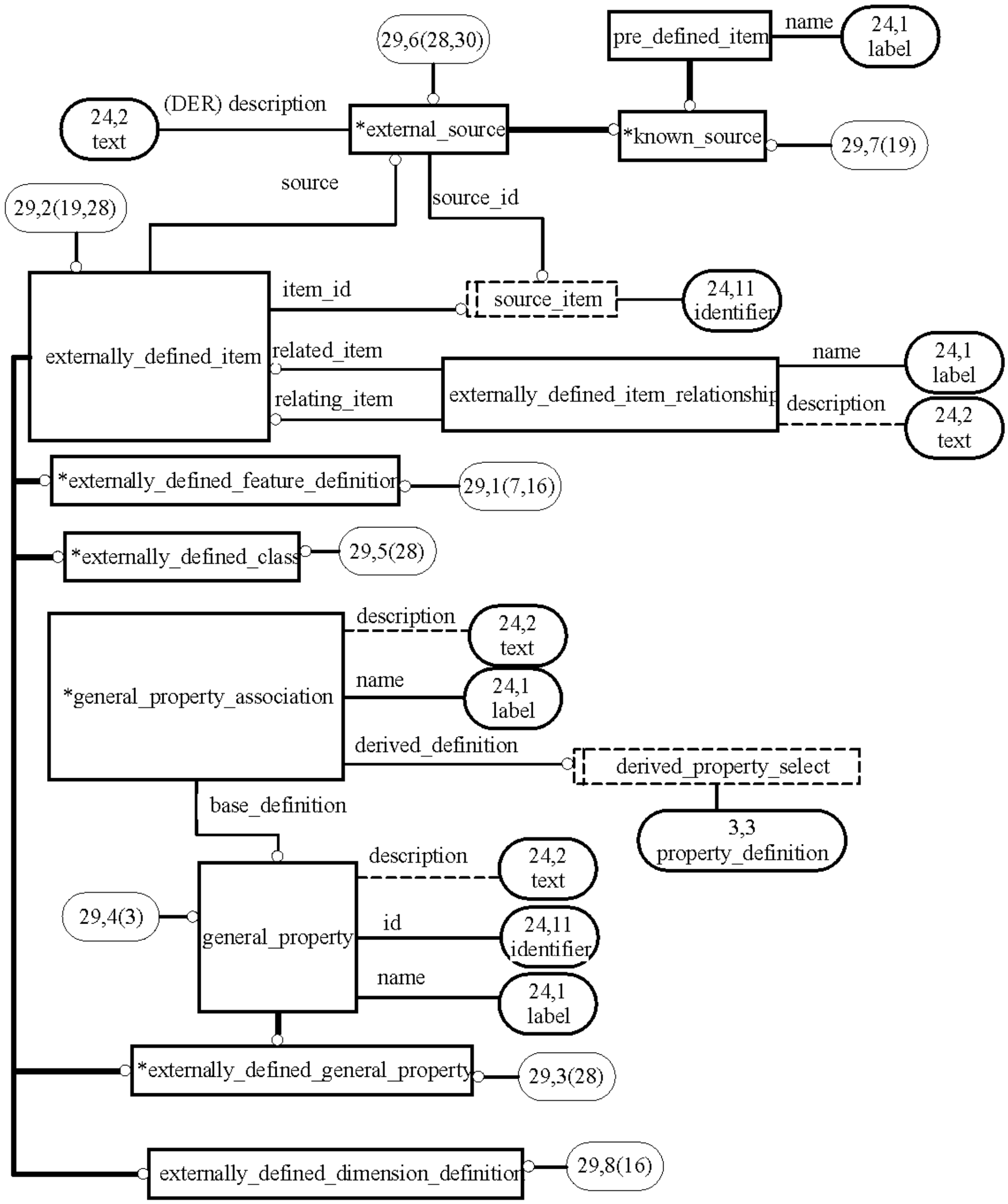


Figure H.29 — externally defined item — AIM EXPRESS-G diagram 29 of 30

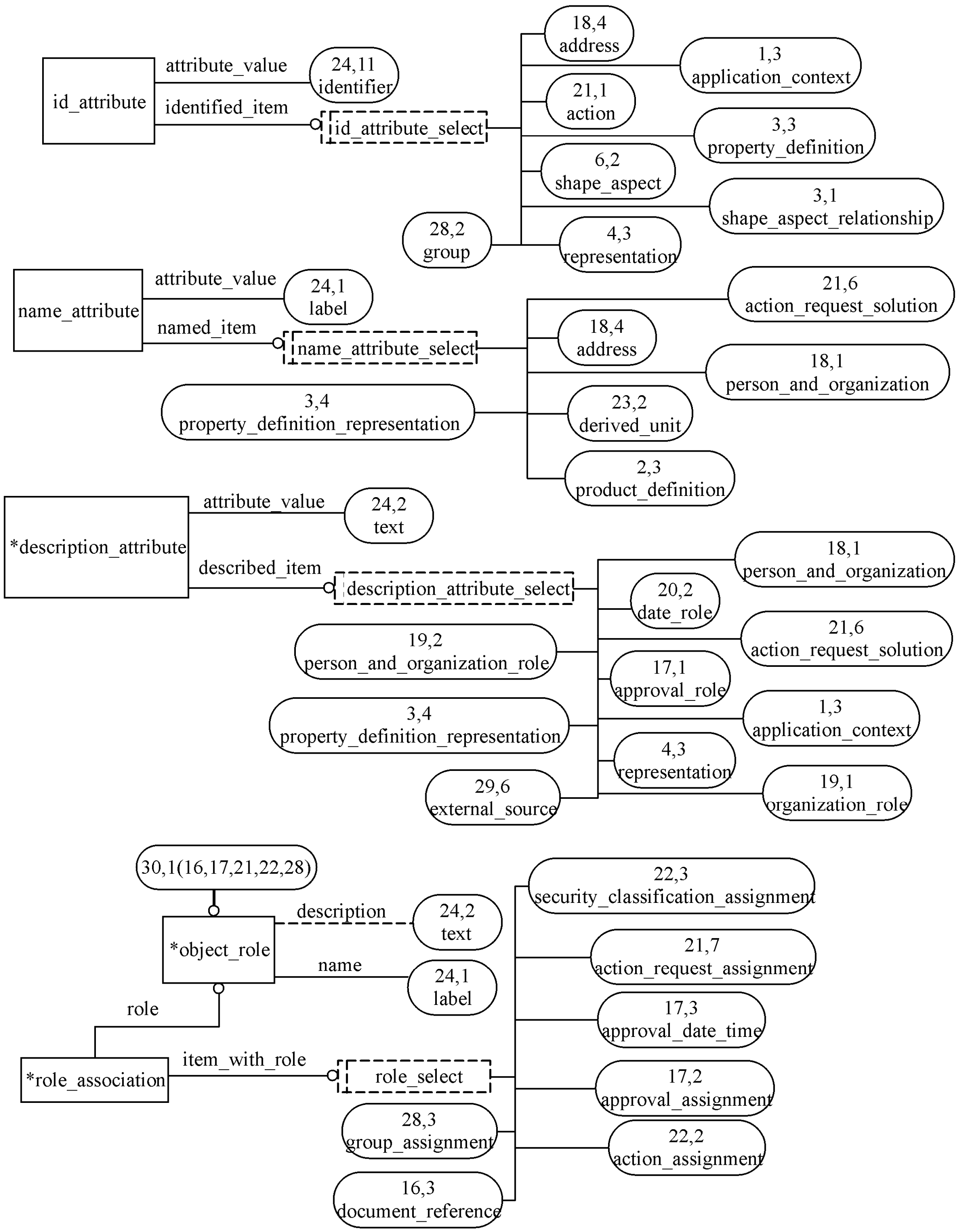


Figure H.30 — attributes — AIM EXPRESS-G diagram 30 of 30

Annex J
(informative)
Computer interpretable listings

A listing of each EXPRESS schema specified in this part of ISO 10303, without comments or other explanatory text, is 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

This annex contains the summary of technical changes for the third edition of this part of ISO 10303. Table L.1 summarizes the changes of the ARM EXPRESS-G model (see Annex G), Application objects (see 4.3), and the Application assertions (see 4.4). Table L.2 summarizes the changes of the AIM EXPRESS short listing (see 5.2).

Table L.1 is organized into 6 columns.

Column 1) Application element: Name of an application element as it appears in the application object definition in 4.2., and the ARM EXPRESS-G model in Annex G.

Column 2) Attribute: Name of an application element attribute, if applicable, as it appears in the application object definition in 4.2., and the ARM EXPRESS-G model in Annex G.

Column 3) Status: Description of the change made in this third edition. Types of changes include: new, modified, or removed application elements or attributes.

Column 4) Upward Compatible: Indicates if the change is or is not upward compatible. 'Yes' indicates the changes is upward compatible, 'no' indicates the changes in not upward compatible.

Column 5) Annex G diagram: The reference to the Annex G diagram that shows the application element or attribute.

Column 6) Comments: General summary of the type of change made to the application element or attribute for the 2nd edition of this part.

Table L.2 is organized into 6 columns.

Column 1) AIM element: Name of an element as it appears in the application object definition in 5.2., AIM EXPRESS expanded listing in Annex A, and the AIM EXPRESS-G model in Annex H.

Column 2) Status: Description of the change made in this third edition. Types of changes include: new, modified, or removed AIM elements, types, rules, or functions.

Column 3) First edition clause: Clause number where the AIM element can be found in the ISO 10303-224:1999 edition.

Column 4) Second edition clause: Clause number where the AIM element can be found in this third edition.

Column 5) Upward Compatible: Indicates if the change is or is not upward compatible. 'Yes' indicates the changes is upward compatible, 'no' indicates the changes in not upward compatible.

Column 6) Comments: General summary of the type of change made to the application element or attribute for the 2nd edition of this part of ISO 10303.

Table K.1 — Summary of 3rd edition ARM changes

Application Entity	Attribute	Status	Upward compatible	Annex G diagram	Comments
Mating_definition	standard_part_fastener	New	Yes	G.2	New attribute
Explicit_base_shape_representation	explicit_shape,				
	user_definied_description	New	Yes	G.7	New attributes
General_pocket	volume_not_removed	New	Yes	G.12	New attribute
Rectangular_closed_pocket	volume_not_removed	New	Yes	G.12	New attribute
Rectangular_open_pocket	volume_not_removed	New	Yes	G.12	New attribute
Recess	volume_not_removed	New	Yes	G.12	New attribute
Planar_face	volume_not_removed	New	Yes	G.15	New attribute
Step	volume_not_removed	New	Yes	G.15	New attribute
Boss		No	Yes	G.17	Now referenced by other entities as a volume not removed
Outer_diameter_to_shoulder	feature_length	New	Yes	G17	New attribute
Gear		New	New	G.21	New entity
Defined_gear		New	New	G.21	New entity
Catalogue_gear		New	New	G.21	New entity
Bevel_gear		New	New	G.21	New entity
Spur_gear		New	New	G.21	New entity
Helical_gear		New	New	G.21	New entity
Externally_defined_representation		New	New	G.21	New entity
Library_part_assignment		New	New	G.21	New entity
Bsu		New	New	G.21	New entity
Class_bsu		New	New	G.21	New entity
Supplier_bsu		New	New	G.21	New entity
Property_bsu		New	New	G.21	New entity
Property_value		New	New	G.21	New entity

Table K.2 — Summary of 3rd edition AIM changes

AIM element	status	2nd edition Clause	3rd edition Clause	Upward compatible	comments
applied_area	moved	5.2.3.1.1	None	Yes	Moved to ISO 10303-522
boss	moved	5.2.3.1.6	None	Yes	Moved to ISO 10303-522
boss_top	moved	5.2.3.1.7	None	Yes	Moved to ISO 10303-522
chamfer	moved	5.2.3.1.8	None	Yes	Moved to ISO 10303-522
chamfer_offset	moved	5.2.3.1.9	None	Yes	Moved to ISO 10303-522
circular_pattern	moved	5.2.3.1.11	None	Yes	Moved to ISO 10303-522
circular_closed_profile	moved	5.2.3.1.10	None	Yes	Moved to ISO 10303-522
closed_path_profile	moved	5.2.3.1.12	None	Yes	Moved to ISO 10303-522
composite_hole	moved	5.2.3.1.13	None	Yes	Moved to ISO 10303-522
compound_feature	moved	5.2.3.1.14	None	Yes	Moved to ISO 10303-522
direction_shape_representation	moved	5.2.3.1.17	None	Yes	Moved to ISO 10303-522
edge_round	moved	5.2.3.1.19	None	Yes	Moved to ISO 10303-522
externally_defined_feature_definition	moved	5.2.3.1.20	None	Yes	Moved to ISO 10303-522
face_shape_representation	moved	5.2.3.1.21	None	Yes	Moved to ISO 10303-522
feature_component_definition	moved	5.2.3.1.30	None	Yes	Moved to ISO 10303-522
feature_component_relationship	moved	5.2.3.1.31	None	Yes	Moved to ISO 10303-522
feature_definition	moved	5.2.3.1.32	None	Yes	Moved to ISO 10303-522
feature_pattern	moved	5.2.3.1.33	None	Yes	Moved to ISO 10303-522
fillet	moved	5.2.3.1.34	None	Yes	Moved to ISO 10303-522
flat_face	moved	5.2.3.1.35	None	Yes	Moved to ISO 10303-522
hole_bottom	moved	5.2.3.1.36	None	Yes	Moved to ISO 10303-522
instanced_feature	moved	5.2.3.1.37	None	Yes	Moved to ISO 10303-522
linear_profile	moved	5.2.3.1.38	None	Yes	Moved to ISO 10303-522
location_shape_representation	moved	5.2.3.1.39	None	Yes	Moved to ISO 10303-522
marking	moved	5.2.3.1.40	None	Yes	Moved to ISO 10303-522
modified_pattern	moved	5.2.3.1.41	None	Yes	Moved to ISO 10303-522
ngon_closed_profile	moved	5.2.3.1.42	None	Yes	Moved to ISO 10303-522
open_path_profile	moved	5.2.3.1.44	None	Yes	Moved to ISO 10303-522
outer_round	moved	5.2.3.1.46	None	Yes	Moved to ISO 10303-522

AIM element	status	2nd edition Clause	3rd edition Clause	Upward compatible	comments
outside_profile	moved	5.2.3.1.47	None	Yes	Moved to ISO 10303-522
partial_circular_profile	moved	5.2.3.1.48	None	Yes	Moved to ISO 10303-522
path_feature_component	moved	5.2.3.1.49	None	Yes	Moved to ISO 10303-522
path_shape_representation	moved	5.2.3.1.50	None	Yes	Moved to ISO 10303-522
pattern_offset_membership	moved	5.2.3.1.51	None	Yes	Moved to ISO 10303-522
pattern_omit_membership	moved	5.2.3.1.52	None	Yes	Moved to ISO 10303-522
planar_shape_representation	moved	5.2.3.1.54	None	Yes	Moved to ISO 10303-522
pocket	moved	5.2.3.1.55	None	Yes	Moved to ISO 10303-522
pocket_bottom	moved	5.2.3.1.56	None	Yes	Moved to ISO 10303-522
profile_floor	moved	5.2.3.1.57	None	Yes	Moved to ISO 10303-522
protrusion	moved	5.2.3.1.58	None	Yes	Moved to ISO 10303-522
rectangular_closed_profile	moved	5.2.3.1.59	None	Yes	Moved to ISO 10303-522
rectangular_pattern	moved	5.2.3.1.60	None	Yes	Moved to ISO 10303-522
removal_volume	moved	5.2.3.1.61	None	Yes	Moved to ISO 10303-522
replicate_feature	moved	5.2.3.1.62	None	Yes	Moved to ISO 10303-522
revolved_profile	moved	5.2.3.1.63	None	Yes	Moved to ISO 10303-522
rib_top	moved	5.2.3.1.64	None	Yes	Moved to ISO 10303-522
rib_top_floor	moved	5.2.3.1.65	None	Yes	Moved to ISO 10303-522
round_hole	moved	5.2.3.1.66	None	Yes	Moved to ISO 10303-522
rounded_end	moved	5.2.3.1.67	None	Yes	Moved to ISO 10303-522
rounded_u_profile	moved	5.2.3.1.68	None	Yes	Moved to ISO 10303-522
shape_defining_relationship	moved	5.2.3.1.69	None	Yes	Moved to ISO 10303-522
shape_representation_with_parameters	moved	5.2.3.1.70	None	Yes	Moved to ISO 10303-522
spherical_cap	moved	5.2.3.1.73	None	Yes	Moved to ISO 10303-522
square_u_profile	moved	5.2.3.1.74	None	Yes	Moved to ISO 10303-522
step	moved	5.2.3.1.75	None	Yes	Moved to ISO 10303-522
slot	moved	5.2.3.1.71	None	Yes	Moved to ISO 10303-522
slot_end	moved	5.2.3.1.72	None	Yes	Moved to ISO 10303-522
taper	moved	5.2.3.1.7	None	Yes	Moved to ISO 10303-522

AIM element	**status**	**2nd edition Clause**	**3rd edition Clause**	**Upward compatible**	**comments**
		6			
tee_profile	moved	5.2.3.1.7 7	None	Yes	Moved to ISO 10303-522
thread	moved	5.2.3.1.7 8	None	Yes	Moved to ISO 10303-522
transition_feature	moved	5.2.3.1.7 9	None	Yes	Moved to ISO 10303-522
turned_knurl	moved	5.2.3.1.8 0	None	Yes	Moved to ISO 10303-522
vee_profile	moved	5.2.3.1.8 1	None	Yes	Moved to ISO 10303-522
applied_classification_assignment	new	5.2.3.3.1	5.2.3.1.	Yes	New entity
applied_external_identification_assig nment	new	5.2.3.3.4	5.2.3.1.	Yes	New entity
class	new	5.2.3.3.8	5.2.3.1.	Yes	New entity
externally_defined_class	new	5.2.3.3.1 2	5.2.3.1.	Yes	New entity
externally_defined_general_property	new	5.2.3.3.1 3	5.2.3.1.	Yes	New entity
externally_defined_representation_wi th_parameters	new	5.2.3.3.1 4	5.2.3.1.	Yes	New entity
known_source	new	5.2.3.3.2 3	5.2.3.1.	Yes	New entity
library_class_version_assignment	new	5.2.3.3.2 4	5.2.3.1.	Yes	New entity
library_property_version_assignment	new	5.2.3.3.2 5	5.2.3.1.	Yes	New entity
mating_definition_requires_externally _defined_representation	new	5.2.3.4.1 6		Yes	New global rule
aim element	status	2nd edition clause	3rd edition clause	Upward compatible	comments
applied_area	moved	5.2.3.1.1	None	Yes	Moved to ISO 10303-522
boss	moved	5.2.3.1.6	None	Yes	Moved to ISO 10303-522

The third edition of this part of ISO 10303 has been updated to incorporate references to externally defined gears and with additional parameters to enable the fundamental information required to capture the specification of the basic geometry. The attribute names, added to allow the third edition of this part of ISO 10303 to represent gears, are identified below and referenced back to the source definitions in ISO 1122-1.

While ISO 1122-1 is already identified as a normative reference within this part of ISO 10303, this table cross-referencing the terms is intended to benefit gear experts who are not already familiar with this part of ISO 10303.

Table K.3 — Summary of 3rd edition changes for gear entity

ARM term	Application object references	reference term	ISO 1122-1 edition reference.	comment
Bevel_gear	4.2.8	bevel gears	1.1.1.5	
Catalogue_gear	4.2.18	-	-	Identifies a gear defined externally.
Defined_gear	4.2.59	-	-	Identifies a gear whose parameters are defined within AP.
face_width	4.2.88.2	facewidth	2.1.1.14	
Helical_bevel_gear	4.2.109	helical bevel gear	1.2.6.8	
Helical_gear	4.2.110	helical gear	1.2.6.3	
internal_or_external_gear	4.2.88.3	external gear internal gear	1.2.2.7 1.2.2.8	identifies internal or external gear form
left_or_right_tooth	4.2.109.1	left-hand teeth right-hand teeth	1.2.6.5 1.2.6.4	
module_or_diameter_pitch	4.2.88.4	diametral pitch module	1.2.1.6 1.2.1.5	Indicator set to "diametral pitch" or "module". The term diametral_pitch already used for turned knurl in AP.
nominal_tooth_depth	4.2.88.5	tooth depth	2.1.3.1	tooth_depth already used for turned knurl in AP.
normal_attribute	4.2.88.6	-	-	The value of "diametral pitch" or "module" key from module_or_-diameter_pitch.
number_of_teeth	4.2.88.7	number of teeth	1.1.2.12	number_of teeth also used for turned knurl in AP, but should not give rise to any confusion.
pitch_angle	4.2.8, identified as a design rather than manufacturing requirement.	pitch angle	3.1.2.2	
profile_shift	4.2.88.8	profile shift	2.1.8.6	
reference_helix_angle	4.2.109.2	helix angle	1.4.1.2	The term helix_angle already used for knurl in AP.
reference_pressure_angle	4.2.88.9	pressure angle	3.1.5.2	
root_angle	4.2.8.1	root angle	3.1.2.6	
root_fillet_radius	4.2.88.10	fillet	1.2.5.5	
shaft_angle	4.2.8, identified as a design rather than manufacturing requirement.	shaft angle	1.1.1.8	

ARM term	Application object references	reference term	ISO 1122-1 edition reference.	comment
Spur_gear	4.2.230	spur gear	1.2.6.1	
Straight_bevel_gear	4.2.233	straight bevel gear	1.2.6.2	
tip_angle	4.2.8.2	tip angle	3.1.2.5	
tip_diameter	4.2.88.11	tip diameter	2.1.1.12	

Bibliography

[1] *IDEF0 Federal Information Processing Standards Publication 183, Integration Definition for Function Modeling (IDEF0)*, FIPS PUB 183, National Institute of Standards and Technology, December 1993.

[2] ANSI Y14.5M:1982. American National Standard, Dimensioning and Tolerancing.

Index

ICS 25.040.40

Price based on 246 pages